蘋果用家月刊

純蘋果電腦用家雜誌　　FEB,1985 VOL,4 NO.21



目錄

APPLE USER MONTHLY FEB., 85 # 21

Pro DOS專欄

再談ProDOS BASIC —— 2

ProDOS ONLINE命令 —— 6

DOS 3.3專欄

Drive 2 Boot 碟 —— 10

實用程式

AMPER GO —— 12

Applesoft BASIC 擴展功能命令

SHAPE TABLE助理 —— 18

Apple soft 視窗 —— 19

程式樂園

JOY PAINT 繪圖軟件 —— 22

從低解象到高解象 —— 25

趣味圖案集

趣味圖案集 —— 30

Apple USER #21 無題 鏡子 視窗示範 奇怪顯示

電視機壞了 正方形 彈彈球 龍的圖 N 角形

小小程式集

魔方陣 —— 35

內海波輪 —— 36

初階篇

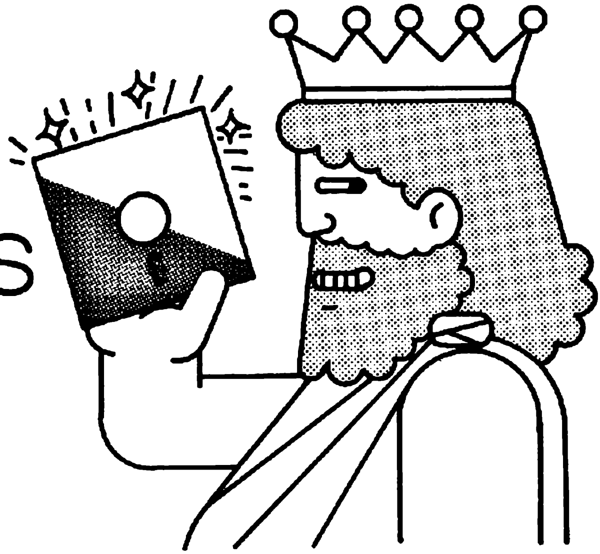
滙編語言入門(上) —— 37

編輯室

蘋果信箱 —— 45

訂閱表 —— 50

再談 ProDOS BASIC



在上一期我們介紹過ProDOS的BASIC·SYSTEM 的指標程式，知道爲甚麽在載入PRODOS 和BASIC·SYSTEM 兩個檔案之後會自動找尋STARTUP 和執行這個程式的各種原因；今次我們將會看看ProDOS BASIC 的$BE00—$BEFF，簡稱爲GP，它的功能與PRODOS 的$BF00—$BFFF頁差不多。

這個GP頁，可以說是整個ProDOS BASIC的中樞，情形就好像MLI是ProDOS 的中樞一樣。在GP地址$BE00—$BEFF 有着由BASIC·SYSTEM設定及使用的重要向量，參數及子程式等，這些功能都隨時可以供BASIC 及滙編語言程序員使用的。

進入ProDOS BASIC

地址$BE00(見程式1)，是一個進入ProDOS BASIC 暖起動進入點的向量位置，在記憶地址的第3頁($300—$3FF)中的兩個向量$3D0及$3D3皆指向地址$BE00 ；ProDOS 與DOS 3·3不同，於

啓動的進入點並非設在BASIC翻譯器（簡稱爲BI）之內。

錯誤訊息

在程式（1）的地址$BE09處。是把程式控制權交給ERROR錯誤訊息去，這是一個由BI負責的主要錯誤訊息指示程式，首先把錯誤碼由累加器A轉送至地址$B—EOF，跟着便進行一些預備的工作。如果ONERR旗號（$D8）是設定的，便沒有錯誤訊息印出，如果ONERR旗號是除下的，則子程式便會執行，它會印出錯誤訊息，並且返回Applesoft。

請注意$BE09的程式並不會返回呼喚程式的部份，但$BE0C却會。在使用$BE09子程式時，必須注意ONERR旗號是否除下，而使用$BE0C時，在表1中載有關於錯誤碼及其相對之訊息的資料，供大家參考。

表1、BASIC翻譯器錯誤碼

(02)	RANGE ERROR
(03)	NO DEVICE CONNECTED
(04)	WRITE-PROTECTED
(05)	END OF DATA
(06)	PATH NOT FOUND
(07)	PATH NOT FOUND
(08)	I/O ERROR
(09)	DISK FULL
(0A)	FILE LOCKED
(0B)	INVALID PARAMETER
(0C)	NO BUFFERS AVAILABLE
(0D)	FILE TYPE MISMATCH
(0E)	PROGRAM TOO LARGE
(0F)	NOT DIRECT COMMAND
(10)	SYNTAX ERROR
(11)	DIRECTORY FULL
(12)	FILE NOT OPEN
(13)	DUPLICATE FILE NAME
(14)	FILE BUSY
(15)	FILE(S) STILL OPEN

執行ProDOS命令

在程式（1）的地址$BE03是BI．GP的最有用之子程式之一，它是一個可以執行ProDO—S BASIC命令之滙編程式。

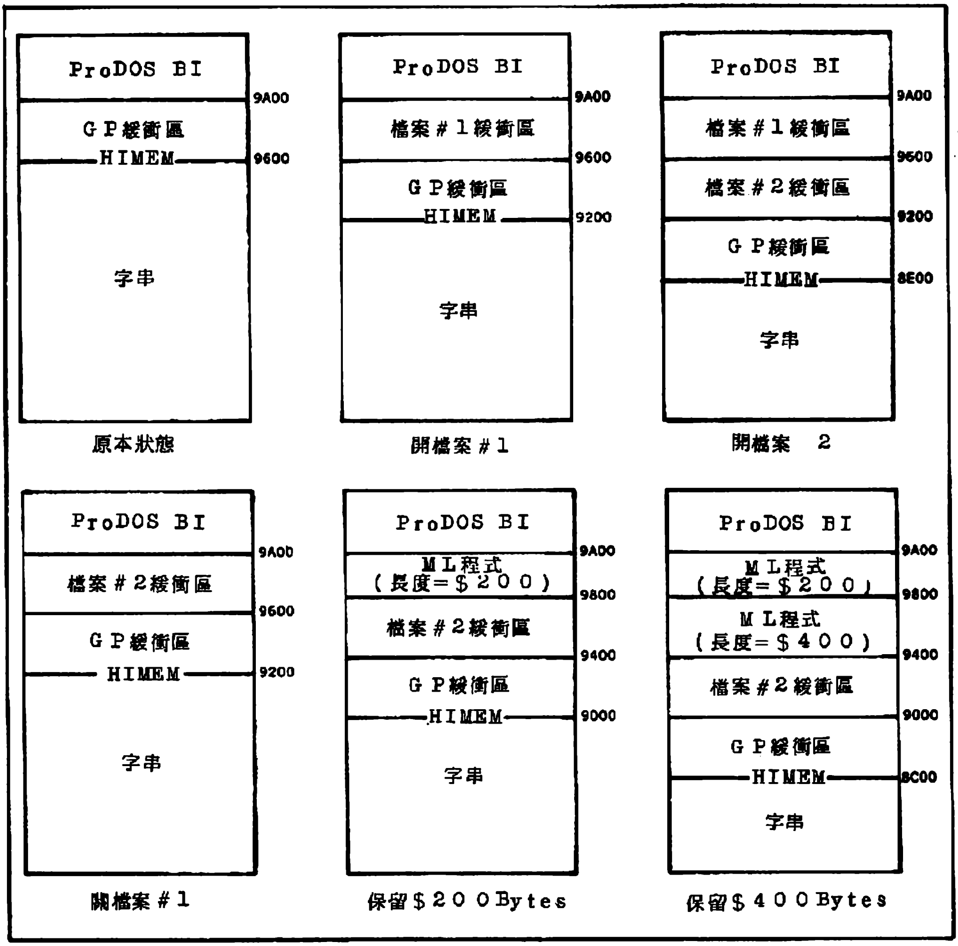
它的使用法十分簡單，只要你把命令的負ASCII碼放在輸入緩衝區（$200）之內，再加上一個<RETURN>，接着CALL這個$BE03，BI便會執行這個命令。

如果在執行時有錯誤訊息出現，進位旗號便會設定爲1，而BI錯誤碼便會存在累加器上。你可以使用$BE09印出錯誤訊息，再返回Applesoft，或者使用$BE0C印出錯誤訊息但返回呼喚其程式之部份。如果在執行$BE03後，進位旗號是0時，這表示命令已成功地執行。

檔案緩衝管理

ProDOS BASIC在保留檔

圖1、檔案緩衝管理



案緩衝區方面的做法，是與DOS 3.3不盡相同的。在DOS3.3時，它自己首先設定了3個檔案緩衝區，它可以利用MAXFILES命令和HIMEM去增加或減少這些檔案緩衝區。而ProDOS 則是由BI所決定。

ProDOS 的檔案緩衝區佔$400bytes 。在設定及執行B—ASIC-SYSTEM檔案時，H—IMEM會被設定爲$9600，而一個1K長的「一般用途緩衝區」（簡稱GPB）是由HIMEM（$9600）至$99FF的，而B

—I則載於$9A00—$BEF—F。GPB是供一些命令，如C—ATALOG，CAT及PREFIX等使用。

在圖1裏你可以見到在開檔案時出現的情形。當第一個檔案（檔案#1）開啓時，其緩衝區便會佔用地址$9600—99FF，即GPB原來佔用的地址，而新的G—PB地址，則順延移至$9200—$95FF，而HIMEM 則設定爲$9200。

如果檔案#1是在開其它檔案之前關掉，則HIMEM 將會設回$9600，而GPB則使用回$9600—99FF地區。

但如果第二個檔案（檔案#2）是在檔案#1關掉前開啓，則會：

1. HIMEM 會設定爲$8E00。
2. GPB會設定於$8E00—$91FF。
3. 檔案#2緩衝區設於$9200—$95FF。
4. 檔案#1仍舊處於$9600—$99FF。

Applesoft 的字串是存於HIMEM 之下的，當HIMEM位置移動時，字串會跟着HIMEM 一起搬移的。

ProDOS BASIC 使用一個簡單的方法，可以在BI和檔案緩衝區之間做成一個空間供機械語程式（ML）使用的。首先把需要空置的地址頁數存於累加器A內，再CALL 地址$BFF5（見程式2）便會做成空位（見圖1）。

如果執行成功的話，進位旗號會被設定爲0，而累加器則存有這個空間地址的首兩個16進數字；假若執行失敗（例如不夠RAM），則進位旗號便會被設定爲1，而累加器則存有錯誤碼。

用地址$BEF5設定的空間，都可以用地址$BEF8（見程式2）的程式取消，它會把所有保留的空間取消，所以如果你有程式存放於BI和檔案緩衝區之間的空間，便要小心使用這個程式了，它會把程式取消。■

程式1

```
BE00-   4C 35 AC    JMP   $AC35
BE03-   4C B4 A6    JMP   $A6B4
BE06-   4C 9E BE    JMP   $BE9E
BE09-   4C 22 9B    JMP   $9B22
BE0C-   4C BF 9F    JMP   $9FBF
BE0F-   00          BRK
```

程式2

```
BEF5-   4C F0 A2    JMP   $A2F0
BEF8-   4C 3C A3    JMP   $A33C
```

ProDOS

ONLINE 命令

大家都知道，Apple Ⅱ本身是沒有磁碟機控制命令的，例如：CATALOG、SAVE、LOAD 等。在ProDOS 裏，這些命令都是附加上去的，而這些命令都是存於一個名爲BASIC.SYSTEM 的檔案內，而Apple Ⅱ是可以直接知道那些是命令。

如果你要加上自己的命令，一般來說，可以使用CALL、USR或者&等，但是若可以直接輸入名字（如CATALOG般），豈不是更加好？

以下要介紹的這個程式，便是能令ProDOS BASIC 命令群中增加一個名爲ONLINE 的命令。

程式簡釋

程式佔地址$2000至$2221，在鍵入程式後，可以按：

BSAVE COMMAND，A$2000，E$2221

儲起來，如果想把這個命令加在ProDOS BASIC上，可以按：

－COMMAND

執行後畫面將沒有甚麼改變，只會再出現游標而已。

現在你只要按下：

ONLINE

便會在畫面出現你現有的磁碟其位置及磁碟名字，例如：

```
]ONLINE
Slot  Drive   Volume Name

 6      1     /USERS.DISK
 6      2     /PRODOS3.3
```

便表示你的Apple Ⅱ，在Slot 6裏有兩部磁碟機，其磁碟的名字分別爲／USERS．DISK 及／P－RODOS 3．3。此外，這個命令亦與其它的ProDOS BASIC命令般可以容許有參數的，以下是其使用示範：

```
]ONLINE,S6
Slot   Drive     Volume Name

  6      1       /USERS.DISK


]ONLINE,D2
Slot   Drive     Volume Name

  6      2       /PRODOS3.3


]ONLINE,S6,D1
Slot   Drive     Volume Name

  6      1       /USERS.DISK


]ONLINE,S6,D2
Slot   Drive     Volume Name

  6      2       /PRODOS3.3
```

其實類似的功能，亦可以在／USERS．DISK 中的FILER 檔案中找到，在VOLUME COMMAN－DS 裏的LIST VOLUME便是有此功能。

讀者可以參考這個程式，作爲自己創造ProDOS BASIC 命令的一個藍本。這個ONLINE 命令亦可以在Apple Ⅱc及Ⅱe的80 column 上工作。■

程式一

```
2000-   A9 02       LDA   #$02
2002-   20 F5 BE    JSR   $BEF5
2005-   B0 4F       BCS   $2056
2007-   48          PHA
2008-   85 43       STA   $43
200A-   AC 07 BE    LDY   $BE07
200D-   AD 08 BE    LDA   $BE08
2010-   8C 3A 21    STY   $213A
2013-   8D 3B 21    STA   $213B
2016-   A9 00       LDA   #$00
2018-   8D 07 BE    STA   $BE07
201B-   38          SEC
201C-   68          PLA
201D-   8D 08 BE    STA   $BE08
2020-   E9 21       SBC   #$21
2022-   85 3E       STA   $3E
2024-   A2 00       LDX   #$00
2026-   A0 00       LDY   #$00
2028-   BD 59 20    LDA   $2059,X
202B-   85 06       STA   $06
202D-   E8          INX
202E-   BD 59 20    LDA   $2059,X
2031-   85 07       STA   $07
2033-   18          CLC
2034-   B1 06       LDA   ($06),Y
2036-   65 3E       ADC   $3E
2038-   91 06       STA   ($06),Y
203A-   E8          INX
203B-   E0 0C       CPX   #$0C
203D-   90 E9       BCC   $2028
```

```
203F-   A0 00       LDY   #$00
2041-   A9 21       LDA   #$21
2043-   84 3C       STY   $3C
2045-   85 3D       STA   $3D
2047-   A0 1F       LDY   #$1F
2049-   A9 22       LDA   #$22
204B-   84 3E       STY   $3E
204D-   85 3F       STA   $3F
204F-   A0 00       LDY   #$00
2051-   84 42       STY   $42
2053-   4C 2C FE    JMP   $FE2C
2056-   4C 09 BE    JMP   $BE09
2059- 06 21 0B 21 11 21 16
2060- 21 2F 21 8D 21 00 00 00
2068- 00 00 00 00 00 00 00 00
2070- 00 00 00 00 00 00 00 00
2078- 00 00 00 00 00 00 00 00
2080- 00 00 00 00 00 00 00 00
2088- 00 00 00 00 00 00 00 00
2090- 00 00 00 00 00 00 00 00
2098- 00 00 00 00 00 00 00 00
20A0- 00 00 00 00 00 00 00 00
20A8- 00 00 00 00 00 00 00 00
20B0- 00 00 00 00 00 00 00 00
20B8- 00 00 00 00 00 00 00 00
20C0- 00 00 00 00 00 00 00 00
20C8- 00 00 00 00 00 00 00 00
20D0- 00 00 00 00 00 00 00 00
20D8- 00 00 00 00 00 00 00 00
20E0- 00 00 00 00 00 00 00 00
20E8- 00 00 00 00 00 00 00 00
20F0- 00 00 00 00 00 00 00 00
20F8- 00 00 00 00 00 00 00 00
2100-   A2 00       LDX   #$00
2102-   A0 00       LDY   #$00
2104-   20 43 21    JSR   $2143
2107-   F0 2F       BEQ   $213B
2109-   D9 3D 21    CMP   $213D,Y
210C-   D0 2A       BNE   $213B
210E-   C8          INY
```

```
210F-   CC 3C 21    CPY   $213C
2112-   90 F0       BCC   $2104
2114-   20 43 21    JSR   $2143
2117-   D0 1F       BNE   $213B
2119-   88          DEY
211A-   8C 52 BE    STY   $BE52
211D-   A9 00       LDA   #$00
211F-   8D 53 BE    STA   $BE53
2122-   A9 80       LDA   #$80
2124-   8D 54 BE    STA   $BE54
2127-   A9 04       LDA   #$04
2129-   8D 55 BE    STA   $BE55
212C-   A0 52       LDY   #$52
212E-   A9 21       LDA   #$21
2130-   8C 50 BE    STY   $BE50
2133-   8D 51 BE    STA   $BE51
2136-   18          CLC
2137-   60          RTS
2138-   38          SEC
2139-   4C 00 00    JMP   $0000
213C- 06 CF CE CC
2140- C9 CE C5
2143-   BD 00 02    LDA   $0200,X
2146-   E8          INX
2147-   C9 A0       CMP   #$A0
2149-   F0 F8       BEQ   $2143
214B-   C9 8D       CMP   #$8D
214D-   F0 02       BEQ   $2151
214F-   C9 AC       CMP   #$AC
2151-   60          RTS
2152-   AD 57 BE    LDA   $BE57
2155-   29 04       AND   #$04
2157-   F0 15       BEQ   $216E
2159-   AD 61 BE    LDA   $BE61
215C-   0A          ASL
215D-   0A          ASL
215E-   0A          ASL
215F-   0A          ASL
2160-   0A          ASL
2161-   8D C7 BE    STA   $BEC7
```

```
2164-   AD 62 BE    LDA   $BE62
2167-   4A          LSR
2168-   4A          LSR
2169-   6E C7 BE    ROR   $BEC7
216C-   D0 03       BNE   $2171
216E-   8D C7 BE    STA   $BEC7
2171-   A0 80       LDY   #$80
2173-   A9 02       LDA   #$02
2175-   8C CB BE    STY   $BECB
2178-   8D C9 BE    STA   $BEC9
217B-   84 06       STY   $06
217D-   85 07       STA   $07
217F-   A9 C5       LDA   #$C5
2181-   20 70 BE    JSR   $BE70
2184-   B0 2E       BCS   $21B4
2186-   20 58 FC    JSR   $FC58
2189-   A2 00       LDX   #$00
218B-   BD 96 21    LDA   $2196,X
218E-   F0 27       BEQ   $21B7
2190-   20 ED FD    JSR   $FDED
2193-   E8          INX
2194-   D0 F5       BNE   $218B
2196- D3 EC
2198- EF F4 A0 A0 A0 C4 F2 E9
21A0- F6 E5 A0 A0 A0 A0 D6 EF
21A8- EC F5 ED E5 A0 CE E1 ED
21B0- E5 8D 8D 00
21B4-   4C 09 BE    JMP   $BE09
21B7-   A0 00       LDY   #$00
21B9-   B1 06       LDA   ($06),Y
21BB-   F0 5E       BEQ   $221B
21BD-   85 42       STA   $42
21BF-   29 0F       AND   #$0F
21C1-   F0 4F       BEQ   $2212
21C3-   8D 20 22    STA   $2220
21C6-   A9 01       LDA   #$01
21C8-   8D 21 22    STA   $2221
21CB-   A5 42       LDA   $42
21CD-   10 03       BPL   $21D2
21CF-   EE 21 22    INC   $2221
21D2-   29 70       AND   #$70
21D4-   4A          LSR
21D5-   4A          LSR
21D6-   4A          LSR
21D7-   4A          LSR
21D8-   48          PHA
21D9-   A2 02       LDX   #$02
21DB-   20 4A F9    JSR   $F94A
21DE-   68          PLA
21DF-   20 E3 FD    JSR   $FDE3
21E2-   A2 06       LDX   #$06
21E4-   20 4A F9    JSR   $F94A
21E7-   AD 21 22    LDA   $2221
21EA-   20 E3 FD    JSR   $FDE3
21ED-   A2 06       LDX   #$06
21EF-   20 4A F9    JSR   $F94A
21F2-   A9 AF       LDA   #$AF
21F4-   20 ED FD    JSR   $FDED
21F7-   E6 06       INC   $06
21F9-   B1 06       LDA   ($06),Y
21FB-   09 80       ORA   #$80
21FD-   20 ED FD    JSR   $FDED
2200-   C8          INY
2201-   CC 20 22    CPY   $2220
2204-   90 F3       BCC   $21F9
2206-   20 8E FD    JSR   $FD8E
2209-   AD 57 BE    LDA   $BE57
220C-   29 04       AND   #$04
220E-   D0 0B       BNE   $221B
2210-   C6 06       DEC   $06
2212-   18          CLC
2213-   A5 06       LDA   $06
2215-   69 10       ADC   #$10
2217-   85 06       STA   $06
2219-   D0 9C       BNE   $21B7
221B-   20 8E FD    JSR   $FD8E
221E-   18          CLC
221F-   60          RTS
2220-   00          BRK
2221-   00          BRK
```

Drive 2 Boot碟

這個程式可以讓你在進行非正式ＤＯＳ（即非ＤＯＳ３．３系統的ＤＯＳ）時，可以簡單地在Ｄ—ＯＳ３．３和這些ＤＯＳ之間互相間換使用。

如果你可以把非正式的ＤＯＳ放在Drive 1而把ＤＯＳ３．３的放在Drive 2，在需要時，可以隨時Boot Drive 1或D—rive 2豈不是十分方便？

Boot Drive 1是Appl—e Ⅱ基本功能，但如何令它可以Boot Drive 2呢？

在此我們提供的這個程式使其可以Boot Drive 2的。程式是寫在１６KRAM卡的BANK1處，不過用家可以把它搬到其它地址處。另外了爲方便使用，我們把它和「&」符號命令結合，而在地址＄９６００至＄９６０Ｂ則有一小程式用作連結之用。

程式的輸入

程式輸入時，可以先放在由地址＄２０００開始，在使用時才由磁碟載入ＲＡＭ卡的＄Ｄ０００地址處，在鍵入（程式1）機械碼後，可按：

BSAVE BOOT，A＄２０００，L＄９Ｂ

把它儲起，接着再鍵入程式（２）的HELLO 程式。爲了方便使用這個系統，用家首先要準備一片空磁碟，放在Drive 1處在鍵入程式以後，按入：

POKE 46855，2

把在地址＄Ｂ７０７的RWTS 程式加以改良，再用命令：

INIT HELLO

在程式（２）儲在新的磁碟上，又把程式（１）鍵在這片磁碟上，即這片磁碟必須擁有此兩個檔案，如要執行此程式，可按：

RUN HELLO

執行後會出現：

D1／D2 BOOT LOADE—D

字樣，如你要boot Drive 2，祇須用命令：

&PR#6，D2

便可，當然你亦可以用：

&PR#6，D1

去啓動Drive 1。 ■

程式一

```
2000-   A9 01      LDA   #$01
2002-   8D 07 B7   STA   $B707
2005-   A2 00      LDX   #$00
2007-   BD 8C D0   LDA   $D08C,X
200A-   9D 00 96   STA   $9600,X
200D-   E8         INX
200E-   E0 0C      CPX   #$0C
2010-   D0 F5      BNE   $2007
2012-   A9 4C      LDA   #$4C
```

```
2014-   8D F5 03    STA   $03F5
2017-   A9 00       LDA   #$00
2019-   8D F6 03    STA   $03F6
201C-   A9 96       LDA   #$96
201E-   8D F7 03    STA   $03F7
2021-   60          RTS
2022-   A2 04       LDX   #$04
2024-   BD 00 02    LDA   $0200,X
2027-   E0 02       CPX   #$02
2029-   F0 16       BEQ   $2041
202B-   DD 00 02    CMP   $0200,X
202E-   D0 0E       BNE   $203E
2030-   CA          DEX
2031-   D0 F1       BNE   $2024
2033-   AD 05 02    LDA   $0205
2036-   C9 31       CMP   #$31
2038-   F0 1A       BEQ   $2054
203A-   C9 32       CMP   #$32
203C-   F0 16       BEQ   $2054
203E-   4C 06 96    JMP   $9606
2041-   C9 31       CMP   #$31
2043-   90 F9       BCC   $203E
2045-   C9 38       CMP   #$38
2047-   B0 F5       BCS   $203E
2049-   69 90       ADC   #$90
204B-   85 FD       STA   $FD
204D-   49 E0       EOR   #$E0
204F-   85 FF       STA   $FF
2051-   CA          DEX
2052-   90 D0       BCC   $2024
2054-   A0 FF       LDY   #$FF
2056-   84 3E       STY   $3E
2058-   C8          INY
2059-   84 3C       STY   $3C
205B-   84 FC       STY   $FC
205D-   84 FE       STY   $FE
205F-   A9 03       LDA   #$03
2061-   85 42       STA   $42
2063-   A5 FD       LDA   $FD
2065-   85 3D       STA   $3D
2067-   85 3F       STA   $3F
2069-   A5 FF       LDA   $FF
206B-   85 43       STA   $43
206D-   20 2C FE    JSR   $FE2C
2070-   A0 02       LDY   #$02
2072-   B9 98 D0    LDA   $D098,Y
2075-   91 FE       STA   ($FE),Y
2077-   88          DEY
2078-   10 F8       BPL   $2072
207A-   A0 39       LDY   #$39
207C-   AD 05 02    LDA   $0205
207F-   18          CLC
2080-   69 59       ADC   #$59
2082-   91 FE       STA   ($FE),Y
2084-   6C FE 00    JMP   ($00FE)
2087- AF
2088- BA 36 2C 44 2C 88 C0 4C
2090- 22 D0 2C 8A C0 4C C9 DE
2098- 2C BA C0
```

程式二

```
110  GOSUB 170
120  POKE  - 16247,0: POKE  - 162
     47,0
130  PRINT  CHR$ (4)"BLOAD BOOT,
     A$D000"
140  POKE 60,0: POKE 61,248: POKE
     62,255: POKE 63,255: POKE 66
     ,0: POKE 67,248
150  CALL 768
160  HOME :A$ = "D1/D2 BOOT LOADE
     D": GOSUB 190: END
170  FOR A = 768 TO 782: READ B: POKE
     A,B: NEXT : RETURN
180  DATA  160,0,32,44,254,44,136
     ,192,32,0,208,44,138,192,96
190  HTAB 20 - ( LEN (A$)) / 2: PRINT
     A$: RETURN
```

AMPERGO—Applesoft BASIC擴展功能命令

Applesoft BASIC 在某一方面來說的確是十分完備，但是比起新推出的BASIC 版本來說，却又似乎差了些，爲此我們決定把它的命令功能強化。

AMPERGO新增強命令包括有GOSUB，GOTO，ON GOTO，ON GOSUB及LIST 幾個命令。

AMPERGO命令使用法

& GOSUB和& GOTO

GOSUB 命令擁有以下三種形式：

1. & GOSUB ＜行號＞
2. & GOSUB ＜標記＞
3. & GOSUB ＜列式＞

所有AMPERGO命令皆以「&」符起首；上面的第1. 項& GOSUB 即等於Applesoft 的GOSUB 功能。第2. 項的& GOSUB ＜標記＞是比較重要的，它把正常呼叫子程式的能力，由行號伸延至標記，而這些標記必須以下形式出現：

&＞＜標記＞

在程式1. 處的BASIC 示範程式，便是使用了& GOSUB ＜標記＞；標記是一串字符，但是這些字符不可以是保留字，即命令名稱，因爲在輸入時Applesoft 會

把它當作命令編譯成1 byte，如果一定要使用便需要加上字符括號「”」了，例如可以用命令：

&GOSUB“FREE MEMOR—Y”

&>“FREE MEMORY”

而如果你的標記是需要有空格的(即上例般)，加上字符括號亦是必須的，當然不加亦可以。

而在使用標記時亦需留心到變數的使用，例如程式中的某一行：

10 A=100：&GOSUB(A)

&GOSUB的功能便等於GOSUB 100的功能，而：

10 A$=“KEES”：&GOSUB(A$)

則會找尋標記爲：

&>KEES

的子程式，在示範程式(1)中，&GOSUB“ERROR”的使用，便是由於有變數ER的使用，所以E—RROR使用字符括號着；而程式(2)則是&GOTO 之示範程式。

在使用ON GOTO或ON GO—SUB 時，參數行可能會是有規律的行號，例如：

ON A GOSUB 100，200，300，400，500

那麽第3. 項的功能便可以將其代替成：

&GOSUB(1000*A)

其實在其它大多數的BASIC都擁有這些功能，而現在只是補Appl—esoft BASIC之不足而已。

&ON GOTO和&ON GOSUB

這兩個命令所使用的參數和&GOTO／&GOSUB一樣是可以爲行號，標記及列式；例如：

100 A$=“JUMP”：A=200：A%=400

110 &ON Y GOSUB (A$)、JUMP、(A)、300、A%

&LIST

至於AMPERGO的最後一個命令便是&LIST，你在這個命令之後必須加上行號，標記或列式，例如&LIST MAJAN等。

如果你想印列出多於一行的話，你便需要在&LIST的參數字符後加上一個逗號，例如：

&LIST MAJAN，

要注意，&LIST並不支援多個參數即：

&LIST <標記1>，<標記2>

則祇會印列&><標記1>之內容。

錯誤訊息

AMPERGO的命令如果使用不當，Apple Ⅱ亦會產生三種錯誤訊息，包括：

Syntax Error

• 在命令上沒有「&」或「>」符號

• 標記的首兩個字符並不足成為 Applesoft 之實變數

• 標記少了字符括號引至錯誤

Illegal Quantitly Error

• 標記長度為零（例如：&GOTO（A$）而A$是空的）

• 列式之值為負數（例如：&GOTO（−1）

Undefined Statment Error

• 沒有所要跳至的標記

• 在標記處沒有「>」符號

程式的輸入

程式（3）便是AMPERGO的主程式，當程式輸入後可以用：

BSAVE AMPERGO，A$9400，L$1C3儲在磁碟上。

程式執行後，會自動設定HIMEM於地址$9000處，及「&」連繫命令，如果你想在你的程式中使用AMPERGO 命令，最好加上BRUN AMPERGO 這個命令，就好像示範程式（1）及（2）。□

程式一

```
100  PRINT  CHR$ (4)"BRUN AMPERGO
     "
110  &  GOSUB INIT
120  &  GOSUB USER INPUT :FDIG =
     DIG
130  &  GOSUB USER INPUT
140  PRINT : PRINT "THE SUM OF ";

150  PRINT FDIG;" AND ";DIG;" IS
     ";FDIG + DIG
160  END
170  &  > USER INPUT
180  PRINT : PRINT "ENTER A DIGIT
     ";
190  &  > DIGIT: GET DIG$
200  CALL CLREOS
210  &  GOSUB CHECK
220  IF  NOT ER THEN  PRINT DIG$:
      RETURN
230  &  GOSUB "ERROR": &  GOTO DI
     GIT
240  &  > CHECK:ER = 0
250  LET DIG =  VAL (DIG$): IF (D
     IG) THEN  RETURN
260  IF DIG$ <  > "0" THEN ER = 1

270  IF DIG$ =  CHR$ (3) THEN  STOP

280  RETURN
290  &  > INIT: TEXT : HOME
300  LET CLREOS =  - 958: RETURN

310  &  > "ERROR"
320  LET HPO =  POS (0) + 1:VPO =
      PEEK (37) + 1
330  HTAB 10: VTAB 24: INVERSE
340  PRINT "NOT A DIGIT, REENTER"
     ;
350  HTAB HPO: VTAB VPO: NORMAL
360  RETURN
```

程式二

```
10  PRINT  CHR$ (4)"BRUN AMPERGO"
20  &  GOTO TEST1
30  &  > TEST1: PRINT  CHR$ (33);
         : &  GOTO TEST2
40  &  > TEST2: PRINT  PEEK (66);
         : &  GOTO TEST1
```

程式三

```
9400-   A9 00      LDA   #$00
9402-   85 73      STA   $73
9404-   85 6F      STA   $6F
9406-   A9 90      LDA   #$90
9408-   85 74      STA   $74
940A-   85 70      STA   $70
940C-   A9 17      LDA   #$17
940E-   8D F6 03   STA   $03F6
9411-   A9 94      LDA   #$94
9413-   8D F7 03   STA   $03F7
9416-   60         RTS
9417-   A2 00      LDX   #$00
9419-   86 FE      STX   $FE
941B-   C9 B4      CMP   #$B4
941D-   D0 17      BNE   $9436
941F-   20 B1 00   JSR   $00B1
9422-   20 F8 E6   JSR   $E6F8
9425-   A8         TAY
9426-   C6 A1      DEC   $A1
9428-   F0 17      BEQ   $9441
942A-   20 B1 00   JSR   $00B1
942D-   F0 06      BEQ   $9435
942F-   C9 2C      CMP   #$2C
9431-   D0 F7      BNE   $942A
9433-   F0 F1      BEQ   $9426
9435-   60         RTS
9436-   C9 BC      CMP   #$BC
9438-   F0 16      BEQ   $9450
943A-   C9 CF      CMP   #$CF
943C-   D0 04      BNE   $9442
943E-   4C 95 D9   JMP   $D995
9441-   98         TYA
9442-   C9 AB      CMP   #$AB
9444-   F0 5F      BEQ   $94A5
9446-   C9 B0      CMP   #$B0
9448-   F0 12      BEQ   $945C
944A-   20 B7 00   JSR   $00B7
944D-   4C C9 DE   JMP   $DEC9
9450-   A0 01      LDY   #$01
9452-   B1 B8      LDA   ($B8),Y
9454-   F0 F4      BEQ   $944A
9456-   C9 3A      CMP   #$3A
9458-   F0 F0      BEQ   $944A
945A-   C6 FE      DEC   $FE
945C-   A9 03      LDA   #$03
945E-   20 D6 D3   JSR   $D3D6
9461-   A5 B9      LDA   $B9
9463-   48         PHA
9464-   A5 B8      LDA   $B8
9466-   48         PHA
9467-   A5 76      LDA   $76
9469-   48         PHA
946A-   A5 75      LDA   $75
946C-   48         PHA
946D-   A9 B0      LDA   #$B0
946F-   48         PHA
9470-   20 A5 94   JSR   $94A5
9473-   24 FE      BIT   $FE
9475-   30 03      BMI   $947A
9477-   4C D2 D7   JMP   $D7D2
947A-   68         PLA
947B-   68         PLA
947C-   85 75      STA   $75
947E-   68         PLA
947F-   85 76      STA   $76
9481-   68         PLA
```

```
9482-   85 B8      STA   $B8
9484-   68         PLA
9485-   85 B9      STA   $B9
9487-   68         PLA
9488-   68         PLA
9489-   20 95 D9   JSR   $D995
948C-   A5 B8      LDA   $B8
948E-   D0 02      BNE   $9492
9490-   C6 B9      DEC   $B9
9492-   C6 B8      DEC   $B8
9494-   20 B7 00   JSR   $00B7
9497-   A8         TAY
9498-   20 B1 00   JSR   $00B1
949B-   C0 2C      CPY   #$2C
949D-   F0 03      BEQ   $94A2
949F-   4C DA D6   JMP   $D6DA
94A2-   4C DA D6   JMP   $D6DA
94A5-   20 B1 00   JSR   $00B1
94A8-   A2 00      LDX   #$00
94AA-   86 FF      STX   $FF
94AC-   B0 03      BCS   $94B1
94AE-   4C 3E D9   JMP   $D93E
94B1-   C9 28      CMP   #$28
94B3-   D0 2D      BNE   $94E2
94B5-   20 B2 DE   JSR   $DEB2
94B8-   24 11      BIT   $11
94BA-   30 13      BMI   $94CF
94BC-   24 A2      BIT   $A2
94BE-   30 09      BMI   $94C9
94C0-   20 52 E7   JSR   $E752
94C3-   4C 41 D9   JMP   $D941
94C6-   4C C9 DE   JMP   $DEC9
94C9-   4C 99 E1   JMP   $E199
94CC-   4C 7C D9   JMP   $D97C
94CF-   20 00 E6   JSR   $E600
94D2-   A0 02      LDY   #$02
94D4-   B1 A0      LDA   ($A0),Y
94D6-   99 B7 00   STA   $00B7,Y
94D9-   88         DEY
94DA-   D0 F8      BNE   $94D4

94DC-   B1 A0      LDA   ($A0),
94DE-   F0 E9      BEQ   $94C9
94E0-   AA         TAX
94E1-   E8         INX
94E2-   86 FA      STX   $FA
94E4-   A5 B8      LDA   $B8
94E6-   85 06      STA   $06
94E8-   A5 B9      LDA   $B9
94EA-   85 07      STA   $07
94EC-   A0 04      LDY   #$04
94EE-   B9 BE 95   LDA   $95BE,
94F1-   99 FB 02   STA   $02FB,
94F4-   88         DEY
94F5-   D0 F7      BNE   $94EE
94F7-   CA         DEX
94F8-   F0 20      BEQ   $951A
94FA-   B1 B8      LDA   ($B8),
94FC-   F0 1C      BEQ   $951A
94FE-   C9 3A      CMP   #$3A
9500-   F0 18      BEQ   $951A
9502-   C9 22      CMP   #$22
9504-   D0 08      BNE   $950E
9506-   E6 B8      INC   $B8
9508-   D0 ED      BNE   $94F7
950A-   E6 B9      INC   $B9
950C-   D0 E9      BNE   $94F7
950E-   C9 2C      CMP   #$2C
9510-   F0 08      BEQ   $951A
9512-   99 FC 02   STA   $02FC,
9515-   C8         INY
9516-   C0 02      CPY   #$02
9518-   D0 DD      BNE   $94F7
951A-   A9 FC      LDA   #$FC
951C-   85 B8      STA   $B8
951E-   A9 02      LDA   #$02
9520-   85 B9      STA   $B9
9522-   20 E3 DF   JSR   $DFE3
9525-   85 A0      STA   $A0
9527-   84 A1      STY   $A1
9529-   A5 B8      LDA   $B8
```

```
952B-   C9 FF      CMP   #$FF
952D-   D0 97      BNE   $94C6
952F-   A0 04      LDY   #$04
9531-   B1 A0      LDA   ($A0),Y
9533-   F0 0E      BEQ   $9543
9535-   85 9C      STA   $9C
9537-   88         DEY
9538-   B1 A0      LDA   ($A0),Y
953A-   85 9B      STA   $9B
953C-   20 81 95   JSR   $9581
953F-   A0 02      LDY   #$02
9541-   B0 13      BCS   $9556
9543-   20 64 95   JSR   $9564
9546-   90 84      BCC   $94CC
9548-   C6 FF      DEC   $FF
954A-   A0 04      LDY   #$04
954C-   A5 9C      LDA   $9C
954E-   91 A0      STA   ($A0),Y
9550-   88         DEY
9551-   A5 9B      LDA   $9B
9553-   91 A0      STA   ($A0),Y
9555-   88         DEY
9556-   B1 9B      LDA   ($9B),Y
9558-   85 75      STA   $75
955A-   85 50      STA   $50
955C-   C8         INY
955D-   B1 9B      LDA   ($9B),Y
955F-   85 76      STA   $76
9561-   85 51      STA   $51
9563-   60         RTS
9564-   A6 67      LDX   $67
9566-   A5 68      LDA   $68
9568-   A0 01      LDY   #$01
956A-   86 9B      STX   $9B
956C-   85 9C      STA   $9C
956E-   B1 9B      LDA   ($9B),Y
9570-   F0 42      BEQ   $95B4
9572-   20 81 95   JSR   $9581
9575-   B0 47      BCS   $95BE
9577-   A0 00      LDY   #$00
9579-   B1 9B      LDA   ($9B),Y
957B-   AA         TAX
957C-   C8         INY
957D-   B1 9B      LDA   ($9B),Y
957F-   D0 E9      BNE   $956A
9581-   A0 04      LDY   #$04
9583-   B1 9B      LDA   ($9B),Y
9585-   C9 AF      CMP   #$AF
9587-   D0 2B      BNE   $95B4
9589-   C8         INY
958A-   B1 9B      LDA   ($9B),Y
958C-   C9 CF      CMP   #$CF
958E-   D0 24      BNE   $95B4
9590-   A5 9B      LDA   $9B
9592-   69 05      ADC   #$05
9594-   85 B8      STA   $B8
9596-   A5 9C      LDA   $9C
9598-   69 00      ADC   #$00
959A-   85 B9      STA   $B9
959C-   A6 FA      LDX   $FA
959E-   A0 FF      LDY   #$FF
95A0-   C8         INY
95A1-   CA         DEX
95A2-   F0 12      BEQ   $95B6
95A4-   B1 06      LDA   ($06),Y
95A6-   F0 0E      BEQ   $95B6
95A8-   C9 3A      CMP   #$3A
95AA-   F0 0A      BEQ   $95B6
95AC-   C9 2C      CMP   #$2C
95AE-   F0 06      BEQ   $95B6
95B0-   D1 B8      CMP   ($B8),Y
95B2-   F0 EC      BEQ   $95A0
95B4-   18         CLC
95B5-   60         RTS
95B6-   20 98 D9   JSR   $D998
95B9-   20 B7 00   JSR   $00B7
95BC-   D0 F6      BNE   $95B4
95BE-   60         RTS
95BF-   20 20 24   JSR   $2420
95C2-   00         BRK
```

SHAPE TABLE 助理

這個工具程式可以助你，以不同的SCALE、HCOLOR 和ROT去觀察磁碟內的SHAPE TABLE檔案中的圖形。

在使用一個表後又可以再使用另外一個。如果你是使用ProDOS而沒有80 column顯示能力的，可以把行號20 最後的：

PRINT D$；"CATALOG"

改為

PRINT D$；"CAT"

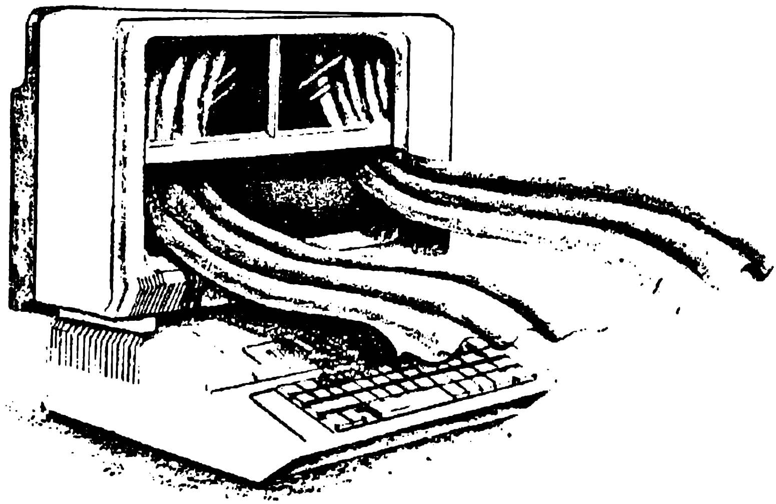
以方便查看磁碟目錄

■

程式一

```
100  HOME : PRINT "HGR SHAPE TABL
     E ASSISTANT":D$ =  CHR$ (4):
      POKE 34,2: PRINT
110  PRINT "PUT YOUR DISK IN THE
     DIRVE, THEN": INPUT "PRESS <
     RETURN>. ";R1$: PRINT D$;"CA
     TALOG"
120  INPUT "FILE NAME OF SHAPE TA
     BLE: ";F$
130  PRINT D$;"BLOAD ";F$: ONERR
      GOTO 220
140 A1% =  PEEK ( - 21902):A2% =
      PEEK ( - 21901):A3% =  PEEK
     (A% + 256 * A2%)
150  POKE 232,A1%: POKE 233,A2%
160  HGR : VTAB 24: PRINT F$;" CO
     NTAINS ";A3%;" SHAPE"; CHR$
     (32 + (A3% > 1) * 51);"."
170  INPUT "        SCALE: ";R2%: INPUT
     "      HCOLOR: ";R3%: INPUT
     "    ROTATION: ";R4%: INPUT
     "SHAPE NUMBER: ";R5%
180  SCALE= R2%: HCOLOR= R3%: ROT=
     R4%
190  DRAW R5% AT 120,65
200  VTAB 24: HTAB 23: INPUT "ANO
     THER? Y/N > ";R6$: IF R6$ =
     "Y" THEN 160
210  VTAB 24: HTAB 21: INPUT "ANO
     THER TABLE ?>";R7$: IF R7$ =
     "Y" THEN  TEXT : RUN
220  POKE 34,0: TEXT : END
```

Applesoft 視窗



你有沒有想過你的Apple Ⅱ可以擁有如LISA、Macintos－h般的視窗功能？當然你可能已擁有Mouse Paint 軟件和老鼠仔，但是Mouse Paint祇是一個用作繪圖的軟件；如果Apple－Soft也可以有視窗，豈不是好得多？

其實要做一個粗簡的視窗並不是一件太困難的事；不過做視窗系統是否有實在的用途呢？如果你想對某些段落進行編輯，視窗便可以專供編輯之用。

例如在AppleSoft BASI－C作編輯時，你需要翻抄某一段落及程式另一段落於一起時，如果你有兩個視窗那麼工作起來便十分簡便。

文字顯示界限

蘋果的文字顯示畫面，其實是可以調校其畫面的。在零頁的地址32至35便是控制着畫面的大小，我們可以用POKE 命令把需要的參數輸入；例如POKE 32，

１０，那麽在畫面顯示時，左面的定位便會由第１０個字符開始，即是左面有１０個空位，至於其它地址的用法見表（１）。

你可以自己嘗試一下利用表中的資料，嘗試一下改變畫面的顯示方式。

程式控制視窗

以下提供兩個程式，一個是用ＡｐｐｌｅＳｏｆｔ　ＢＡＳＩＣ寫成的示範程式（１），它示範了以程式控制的視窗效果，各蘋果用家可以從其中得到靈感、創出自己的視窗程式。

至於另一個程式（２），則是一個機械語言程式，它把畫面分成上下兩部份，你可以用指令ＣＡＬＬ７７１進入上半部畫面，而用ＣＡＬＬ７９４進入下半部畫面，衹要用命令ＴＥＸＴ　便可返回正常的狀態。

而地址７７０是存着分割畫面的比例數目，你可以利用ＰＯＫＥ７７０，ｎ　作出適當的更改。

程式的儲存

ＡｐｐｌｅＳｏｆｔ　ＢＡＳＩＣ的程式（１）可以：

ＳＡＶＥ　ＤＥＭＯ

儲起來，而機械語部份則可以用：

ＢＳＡＶＥ　ＷＩＮＤＯＷＳ，Ａ＄３００，Ｌ＄４０

儲存起來。 ■

表１　控制文字畫面地址及其功能

地址	功能	入數範圍
３２	左邊定位	０—３９
３３	視窗濶度	１—４０
３４	頂邊定位	０—２３
３５	底邊定位	０—２３

程式一

```
100  DIM WS$(24)
110  HOME : PRINT : PRINT  TAB( 1
     1)"WINDOWS DEMONSTRATIONS"
120  PRINT : PRINT  TAB( 15)"BY A
     PPLE USER": PRINT : PRINT
130  FOR J = 1 TO 8
140  FOR K = 1 TO 40: PRINT  CHR$
     (64 + 26 *  RND (1));: NEXT
     K: PRINT
150  NEXT J
160  VTAB 23: PRINT "PRESS ANY KE
     Y TO HALT"
170 WL = 12:WT = 10:WW = 10:WB =
     5: GOSUB 350
180  PRINT : PRINT "GO IT?"
190  GOSUB 340
200  GOSUB 480
210  IF  PEEK (49152) > 128 THEN
      TEXT : HOME : END
220 WL = 5:WT = 1:WW = 25:WB = 7
230  GOSUB 350
240  VTAB WT + 2: HTAB 4: PRINT "
     NOTICE THE TEXT IS": HTAB 4:
      PRINT "RESTORED CORRECTLY."
```

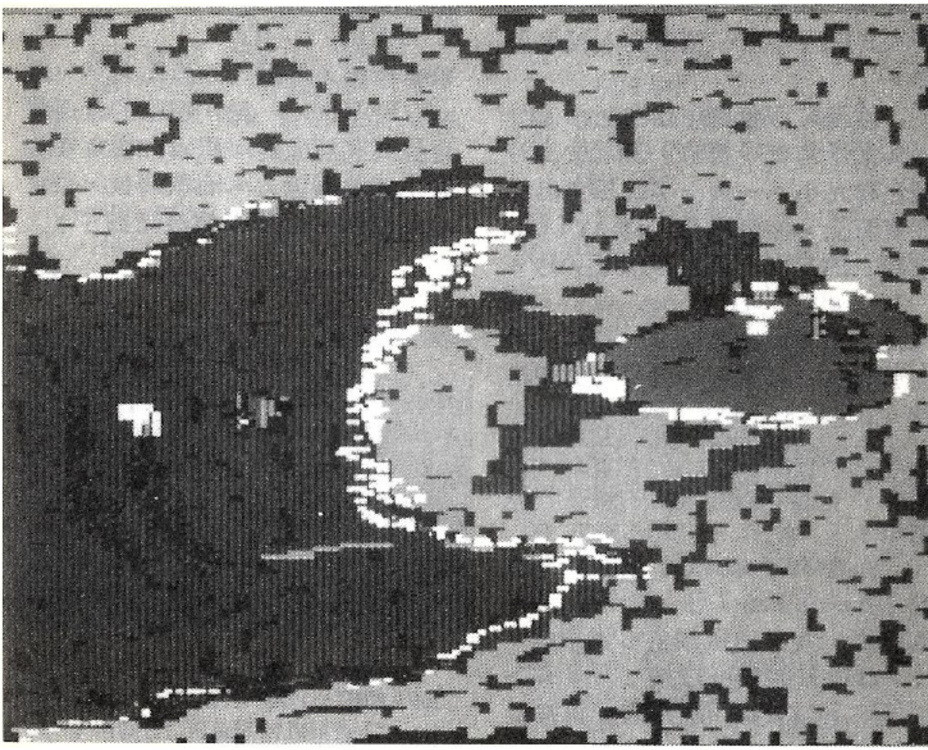
```
250  GOSUB 340
260  GOSUB 480
270  IF  PEEK (49152) > 128 THEN
      TEXT : HOME : END
280 WT = 10:WB = 10: GOSUB 350
290  FOR J = 1 TO 25: PRINT "  ";
     J,J * J: NEXT J
300  PRINT : PRINT "SCROLLING IS
     AUTOMATIC"
310  GOSUB 340: GOSUB 480
320  IF  PEEK (49152) > 128 THEN
      TEXT : HOME : END
330  GOTO 170
340  FOR J = 1 TO 1500: NEXT : RETURN

350 WA = 1024 + 128 * (WT - 1 - 8
      *  INT ((WT - 1) / 8)) + 40
      *  INT (WT / 8.5)
360 WS = WA
370  FOR WJ = WT TO WT + WB - 1:W
     S$(WJ) = ""
380  FOR WK = 1 TO WW:WS$(WJ) = W
     S$(WJ) +  CHR$ ( PEEK (WA +
     WL + WK - 1)): NEXT WK
390  POKE WA + WL,32: POKE WA + W
     L + WW - 1,32
400 WA = WA + 128: IF WA = 2088 THEN
     WA = 1104
410  IF WA = 2048 THEN WA = 1064
420  NEXT WJ
430  FOR WJ = 1 TO WW: POKE WS +
     WL + WJ - 1,32: POKE WA - 12
     8 + 984 * (WA = 1064 OR WA =
     1104) + WL + WJ - 1,32: NEXT
     WJ
440  POKE 32,WL + 1: POKE 33,WW -
     2
450  POKE 34,WT: POKE 35,WT + WB -
     2
460  HOME
470  RETURN
480  POKE 32,0: POKE 33,40
490  POKE 34,0: POKE 35,24
500  FOR WJ = WT TO WT + WB - 1: VTAB
     WJ: HTAB WL + 1: PRINT WS$(W
     J): NEXT WJ
510  RETURN
```

程式二

```
0300- 17 13 0C
0303-   A5 25       LDA   $25
0305-   8D 01 03    STA   $0301
0308-   AD 00 03    LDA   $0300
030B-   85 25       STA   $25
030D-   A9 00       LDA   #$00
030F-   85 22       STA   $22
0311-   AD 02 03    LDA   $0302
0314-   85 23       STA   $23
0316-   20 34 03    JSR   $0334
0319-   60          RTS
031A-   A5 25       LDA   $25
031C-   8D 00 03    STA   $0300
031F-   AD 01 03    LDA   $0301
0322-   85 25       STA   $25
0324-   AD 02 03    LDA   $0302
0327-   18          CLC
0328-   69 01       ADC   #$01
032A-   85 22       STA   $22
032C-   A9 18       LDA   #$18
032E-   85 23       STA   $23
0330-   20 34 03    JSR   $0334
0333-   60          RTS
0334-   AD 02 03    LDA   $0302
0337-   20 24 FC    JSR   $FC24
033A-   A2 27       LDX   #$27
033C-   A9 BD       LDA   #$BD
033E-   20 F0 FD    JSR   $FDF0
0341-   CA          DEX
0342-   D0 FA       BNE   $033E
0344-   60          RTS
```

JoyPAINT 繪圖軟件



這是一個結構簡單的繪圖軟件，原意是供小孩子們使用的，所以在結構上十分簡單。

整個程式是用Applesoft BASIC 寫成的；在程式中行號500～520所載的，是利用命令POKE 儲入的Shape Table 數據，而在行號130，是用作檢查以上的輸入工作。

在螢幕上繪圖

使用這個軟件，祇需鍵入RUN <RETURN> 便可，它首先會出現一個選擇表，包括以下的項目：

<1> LOAD A PICTURE
<2> SAVE A PICTURE
<3> START OR QUIT

如果你沒有磁碟機，便可以忽略第<1 >和<2 >項，直接按下鍵「3」，這時會問你選用繪圖與否，如果按下「Q」表示不用，如果按下<RETURN> 便會進入繪圖世界。

繪圖畫面會在一個白色框之內，其左下角有一標誌，表示繪出的顏色和大小。一個十字遊標會出現在畫面上，你可以用Joystick控制它的方向，祇要你按下 Joys－tick上「bottom 0」，便可以在游標處繪上顏色，而繪上的方塊大小，可以利用鍵盤上的空格鍵(SPACE BAR)去改變其大小，大小改變的情況，可以在左下角處看到，共有三種不同的大小可供選用。

而在Joystick 上的另一個鍵「bottom 1」，是用作顏色的選擇，共有四種顏色可供選擇。

如果玩完後可以按<ESC>鍵返回JOYMENU選擇表，再按<3>及<Q>返回 BASIC 。

圖案的存取

在 JoyMENU 選擇表的第<2>項目，可以讓你把剛畫好的圖案儲存在磁碟上，當你在選擇表處按下<2>字，便會出現：

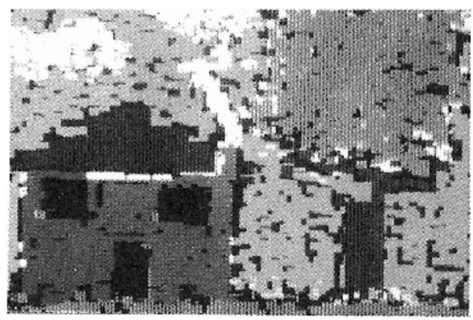
FILENAME :

請你輸入要儲存圖案的檔案名字，再按<RETURN> 便可，在儲存工作完成後會返回選擇表。

如果你要把儲存在磁碟上的圖案取回，可以在選擇表處按<1>，它同樣會出現

FILE NAME :

祇要你鍵入想載入的檔案名字，按<RETURN> 鍵，電腦的畫面便會進入高解像圖；跟着你便會看到圖案的載入過程，而載入後，便會出現游標，你可以修改此圖了。□



```
100  HOME : VTAB (2): HTAB (15): PRINT
     "JoyPAINT": SCALE= 1: ROT= 0

110 EP = .30:BR = 2:XS = 0:XL = 5
     0:YL = 50:CL = 3:SU = 0
120  FOR J = 1 TO 33: READ HX: POKE
     16383 + J,HX:SU = SU + HX: NEXT
     J
130  IF SU < > 1812 THEN  TEXT :
      HOME : PRINT "TYPO IN DATA
     LINES": END
140  VTAB (10): HTAB (10): PRINT
     "<1>  LOAD A PICTURE"
150  VTAB (12): HTAB (10): PRINT
```

```
    "<2>  SAVE A PICTURE"
160  VTAB (14): HTAB (10): PRINT
    "<3>  START OR QUIT"
170 A =  PEEK ( - 16384)
190  IF A = 177 THEN  GOSUB 530: GOSUB
    550: PRINT  CHR$ (4);"BLOAD"
    ;G$;",A$2000": GOSUB 270
200  IF A = 178 THEN  GOSUB 530: PRINT
     CHR$ (4);"BSAVE";G$;",A$200
    0,L$2000": HOME : VTAB (2): HTAB
    (15): PRINT "JoyMENU": GOTO
    140
210  IF A = 179 THEN  VTAB (20): PRINT
    "PRESS <RETURN> TO START, OR
     PRESS <Q> TO QUIT": GOTO 56
    0
220  GOTO 170
230  GOSUB 550
240  HCOLOR= 3: HPLOT 0,0 TO 279,
    0 TO 279,191 TO 0,191 TO 0,0

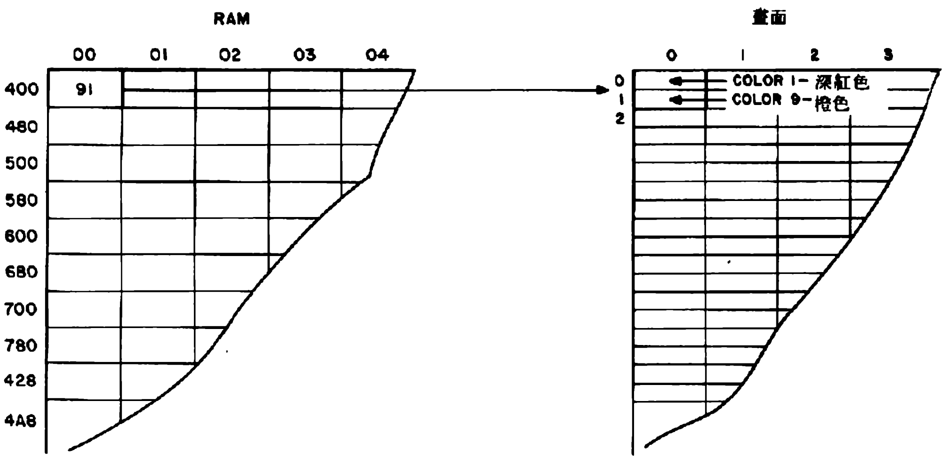
250  GOSUB 370
260  HCOLOR= CL: XDRAW 1 AT XL,YL

270 X = 279 *  PDL (0) / 256:X =
    EP * XL + (1 - EP) * X:X = 2
     *  INT (X / 2) + XS
280  XDRAW 1 AT XL,YL
290 Y = 191 *  PDL (1) / 256:Y =
    EP * YL + (1 - EP) * Y
300  IF  PEEK ( - 16287) > 127 THEN
     DRAW BR AT X,Y
310  XDRAW 1 AT X,Y
320 XL = X:YL = Y
330  IF  PEEK ( - 16286) > 127 THEN
     GOSUB 370
340  IF  PEEK ( - 16384) = 155 THEN
     TEXT : HOME : VTAB (2): HTAB
    (15): PRINT "JoyMENU": GOTO
    140
350  IF  PEEK ( - 16384) = 160 THEN
     GOSUB 450
360  GOTO 270
370  IF CL = 0 AND XS = 0 THEN XS
     = 1: GOTO 440
380  IF CL = 0 THEN CL = 3:XS = 1
    : GOTO 440
390  IF CL = 3 THEN 420
400  IF XS = 0 THEN CL = 0: GOTO
    440
410 XS = 0: GOTO 440
420  IF XS = 1 THEN XS = 0: GOTO
    440
430 XS = 1:CL = 7: GOTO 440
440  GOSUB 470: RETURN
450 BR = BR + 1: IF BR > 4 THEN B
    R = 2
460  GOSUB 470: RETURN
470  HCOLOR= 0: DRAW 2 AT 7,185: DRAW
    2 AT 6,185: HCOLOR= CL
480  DRAW BR AT 6 + XS,185
490  POKE  - 16368,0: RETURN
500  DATA  4,0,10,0,20,0,28,0,31,
    0,36,149
510  DATA  173,219,246,24,8,56,7,
    0,36,108,54,110
520  DATA  36,36,0,255,108,54,0,4
    ,0
530  VTAB (20): HTAB (9): PRINT "
                                 "
540  VTAB (20): HTAB (9): INPUT "
    FILENAME: ";G$: RETURN
550  POKE 232,0: POKE 233,64: HGR
    : POKE  - 16302,0: RETURN
560 A =  PEEK ( - 16384): IF A <
     > 141 AND A <  > 209 THEN 5
    60
570  IF A = 209 THEN  TEXT : HOME
    : END
580  GOTO 230
```

從低解象到高解象

圖（１）低解像圖ＲＡＭ與畫面組織

1 BYTE RAM = 2　畫面像素



要做一個低解象圖案總比高解象爲簡單及容易，如果你在低解象圖裏畫了一個圖案，如果想用打印機印列出來便可能有些麻煩，因爲大部份的打印機介面程式，都只用作印列高解象圖案。

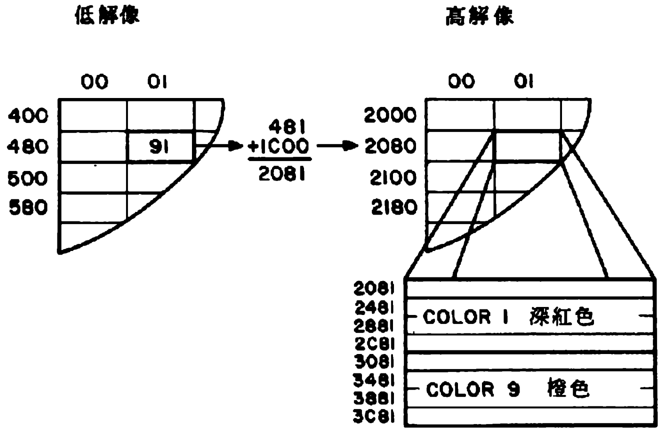
現在我們提供一個可以把低解象圖案轉變爲高解象圖案的程式，使你可以通過打印機把它印列出來。

基本知識

首先讓我們重溫一些已知的資料，它是製作這個程式必須先知的條件。

- 低解象圖畫面佔９６０bytes（４０行×２４列），佔地址＄４００至＄７ＦＦ（符號＄表示１６進數）。
- 每個byte可以控制低解象畫面的兩個像素（Ｐｉｘｅｌ），這兩個象素是上下並列的，這使到低解象的解象度，可達４０×４８個象素。
- 高解象圖畫面佔７６８０byte－s，是8倍於低解象圖案畫面

圖（2）低解像圖至高解像圖RAM地址之關係



（40行×192列）。高解象圖案第1頁佔地址＄2000至＄3FFF。我們把情況看成低解象的一個byte等於8個高解象的bytes，或者一個低解象的象素等於4個高解象圖的象素。

• 把任何低解象圖的象素地址加上＄1C00便等於高解象圖第1頁的相對象素位置的地址。這地址是指象素方塊的最高一個位置（如果加上＄3C00便是相對於高解象圖第2頁的象素）。

低解象圖及其畫面組織

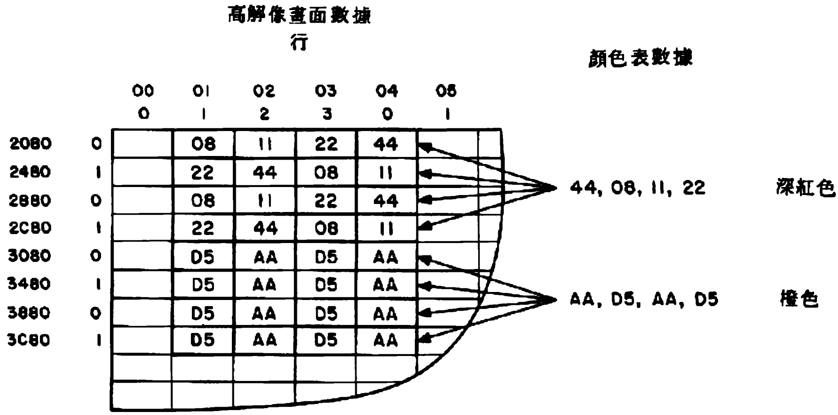
假設數據＄91是儲存於記憶地址＄400處，即低解象圖畫面的左上角（見圖（1））。每個b－yte 的4個bit 的RAM都可以儲存由0至15（＄0－＄F）之數據，而這16個數目即相應對於Apple Ⅱ的16個低解象圖顏色數據（見表（1））。

在圖（1）裏，byte ＄91相應着兩個象素的顏色，橙色（9）和深紅色（1），而較低的4個bit （即1）表示畫面上方的象素，而較高的4個bit（即9）表示畫面下方的象素，即深紅色出現於坐標（0，0）位置而橙色則在（0，1）位置。

低解象及高解象RAM定址

在圖（2）中，我們把在低解象圖地址＄481加上＄1C00，把它變成高解象圖地址＄2081，即得出8個垂直象素的最高之一的地址，在圖（2）裏我們可以看到其它7個象素的地址，它們之間剛好相差＄400。

圖(3)高解像圖畫面數據及其相對於顏色表之數據



表(1)低解像顏色表

數字 十進制	16 進制	顏色
0	$0	黑
1	$1	深紅
2	$2	深藍
3	$3	紫
4	$4	深綠
5	$5	灰1
6	$6	藍
7	$7	淺藍
8	$8	棕
9	$9	橙
10	$A	灰2
11	$B	粉紅
12	$C	淺綠
13	$D	黃
14	$E	水
15	$F	白

高解象彩色描繪

在計算了其相對位置後,你便需要決定哪一個顏色表的byte要描繪了(在程式(1)中地址$382至$3C1是載有顏色表),在正常情況之下,如果你需要在某一個象素處顯示一種顏色,你便需要用2個byte的記憶,例如你想顯示一列藍色象素,便需要把數據$D5和$AA以雙單的行數形式存於RAM內。

在Applesoft ROM內,有關高解象度位置及描點程式部份是位於地址$F411處;如果你把單雙序弄錯,即互換了,那麼你便會得出相反的顏色,如$D5在單行而$AA在雙行,便會出現橙色的一點了。

我們的程式有部份工作是代替了ROM程式的功能這是爲了想盡量在把圖案變爲高解象度時,保留所有顏色的顯示。

在圖(3),顯示於地址$2081的象素,爲了要擁有低解象顏色的效果,便把象素改爲由8b

—yte 控制。

程式的使用

程式(1)是本程式的整個部份，在輸入後，便用命令：

BSAVE LO·HI，A$300，L$C2

儲起要執行程式時，祇需鍵入：

BRUN LO·HI 或：

BLOAD LO·HI再CALL 768或；

—LO·HI(ProDOS 時)

如果你想看看那已變換的高解象圖案，可按：

POKE 49239，0

或者可以按：

POKE 49238，0

再看看低解象的圖案。

如果你想把高解象圖儲存起來，可以用：

BSAVE PICTURE，A$2000，L$2000

等或以其它檔案名字儲存起來。

示範程式

在程式(2)裏是一個由B—ASIC寫成的示範程式，它首先會在低解象圖案畫面處畫上一個蘋果，然後利用程式(1)把它變成高解象圖。

■

主程式

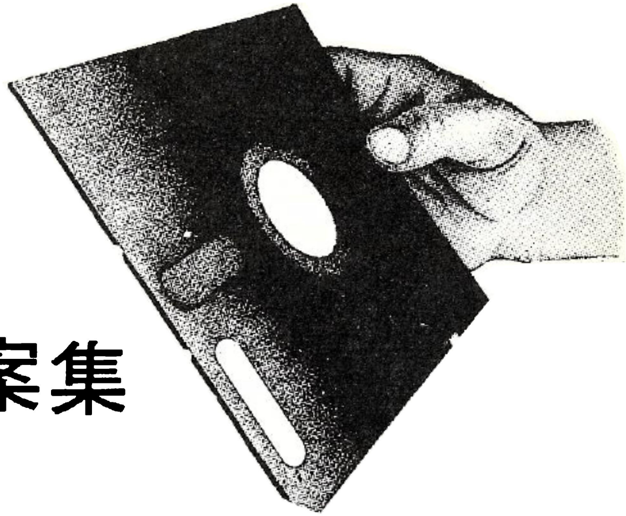
```
0300-   A9 00      LDA   #$00
0302-   85 00      STA   $00
0304-   A8         TAY
0305-   A9 04      LDA   #$04
0307-   85 01      STA   $01
0309-   A9 00      LDA   #$00
030B-   85 06      STA   $06
030D-   A5 00      LDA   $00
030F-   85 02      STA   $02
0311-   18         CLC
0312-   A5 01      LDA   $01
0314-   69 1C      ADC   #$1C
0316-   85 03      STA   $03
0318-   A5 02      LDA   $02
031A-   29 03      AND   #$03
031C-   85 06      STA   $06
031E-   B1 00      LDA   ($00),Y
0320-   48         PHA
0321-   29 0F      AND   #$0F
0323-   20 47 03   JSR   $0347
0326-   20 64 03   JSR   $0364
0329-   68         PLA
032A-   29 F0      AND   #$F0
032C-   18         CLC
032D-   6A         ROR
032E-   6A         ROR
032F-   6A         ROR
0330-   6A         ROR
0331-   20 47 03   JSR   $0347
0334-   20 64 03   JSR   $0364
0337-   E6 00      INC   $00
0339-   D0 08      BNE   $0343
033B-   E6 01      INC   $01
033D-   A5 01      LDA   $01
033F-   C9 08      CMP   #$08
0341-   F0 03      BEQ   $0346
0343-   4C 09 03   JMP   $0309
```

```
0346-   60         RTS
0347-   0A         ASL
0348-   0A         ASL
0349-   18         CLC
034A-   65 06      ADC   $06
034C-   AA         TAX
034D-   BD 82 03   LDA   $0382,X
0350-   85 04      STA   $04
0352-   A5 06      LDA   $06
0354-   C9 02      CMP   #$02
0356-   B0 04      BCS   $035C
0358-   E8         INX
0359-   E8         INX
035A-   D0 02      BNE   $035E
035C-   CA         DEX
035D-   CA         DEX
035E-   BD 82 03   LDA   $0382,X
0361-   85 05      STA   $05
0363-   60         RTS
0364-   A2 00      LDX   #$00
0366-   A5 04      LDA   $04
0368-   91 02      STA   ($02),Y
036A-   18         CLC
036B-   A5 03      LDA   $03
036D-   69 04      ADC   #$04
036F-   85 03      STA   $03
0371-   A5 05      LDA   $05
0373-   91 02      STA   ($02),Y
0375-   18         CLC
0376-   A5 03      LDA   $03
0378-   69 04      ADC   #$04
037A-   85 03      STA   $03
037C-   E8         INX
037D-   E0 02      CPX   #$02
037F-   90 E5      BCC   $0366
0381-   60         RTS

0382- 00 00 00 00 44 08
0388- 11 22 91 A2 C4 88 55 2A
0390- 55 2A 08 11 22 44 33 66
0398- 4C 19 D5 AA D5 AA DD BB
03A0- F7 EE 88 91 A2 C4 AA D5
03A8- AA D5 66 4C 19 33 77 6E
03B0- 5D 3B 2A 55 2A 55 EE DD
03B8- BB F7 3B 77 6E 5D FF FF
03C0- FF FF
```

程式二

```
100  TEXT : HOME
110  PRINT  CHR$ (4)"BLOAD LO;HI"
     : GR
120  COLOR= 12: VLIN 9,10 AT 33: VLIN
     10,11 AT 32: VLIN 11,12 AT 3
     1
130  HLIN 23,38 AT 16: HLIN 24,37
      AT 15: HLIN 24,37 AT 14
140  HLIN 25,29 AT 13: HLIN 32,36
      AT 13: HLIN 27,28 AT 12: HLIN
     33,34 AT 12
150  COLOR= 13: HLIN 23,38 AT 17:
      HLIN 23,37 AT 18: HLIN 23,3
     6 AT 19
160  COLOR= 9: FOR Y = 20 TO 22: HLIN
     23,35 AT Y: NEXT
170  COLOR= 1: HLIN 23,36 AT 23: HLIN
     23,37 AT 24: HLIN 24,38 AT 2
     5
180  COLOR= 3: HLIN 24,38 AT 26: HLIN
     24,38 AT 27: HLIN 25,37 AT 2
     8
190  COLOR= 2: HLIN 26,36 AT 29: HLIN
     26,36 AT 30: HLIN 28,34 AT 3
     1: HLIN 29,30 AT 32: HLIN 32
     ,33 AT 32
200  CALL 768
210  POKE 49239,0: POKE 49168,0
```



趣味圖案集

要在蘋果上做出好的圖案效果，除了要明白其圖案結構，命令的用法和原理外，最佳的辦法莫如親自嘗試製做一些程式。

在趣味圖案集裏，我們提供了一些示範的程式，包括低解像及高解像度的圖案，希望各用家可以從這些簡短的程式中，得到一些幫助，或者有部份程式可以用作你要設計的程式中的子程式呢！

低解像圖案程式

Apple USER #21

程式（１）最適宜用作程式的開始，例如遊戲程式的「片頭」等，其中的行號１１０，變數Ａ＄的內容可以是其它的文字。

無題

當你按下程式（２）後，用Ｒ－ＵＮ執行後你會發現畫面會被清除，但游標又再次出來，甚麼也沒有做，不要以爲程式沒有執行，祇要你按下鍵，便會出現行號１１０變數Ａ＄之內容，小心被騙。

鏡子

這個程式（３），會使你的顯

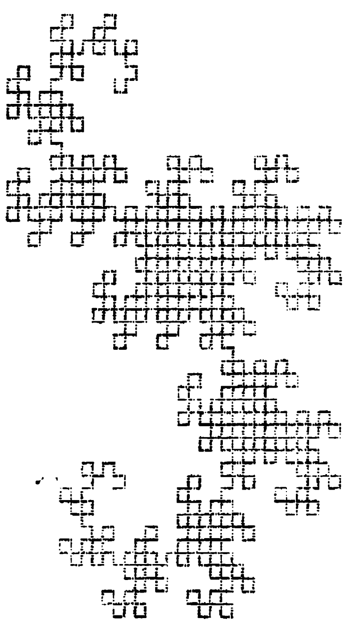
示幕中的畫面，左右反轉，就好像一面鏡子般；其中使用到一些低解像度圖案的命令，值得參考。

視窗示範

這是一個文字顯示幕的視窗示範，程式（４）其中的基本技巧和理論，可以參考本期的「Apple－soft 視窗」一文。

奇怪顯示

程式（５）的效果是無意中發現的，你可以嘗試改變行號１２０的內容，例如改爲Ｘ＝１，Ｘ＝２５５或者其它的命令、述句等，將會有不同的效果。



高解像圖案程式

電視機壞了

程式（６）提供一個類似電視機壞了時的畫面般的效果。

正方形

利用正方形不同旋轉的角度和大小，砌成不同的圖案，這是程式（７）的示範。

彈彈球

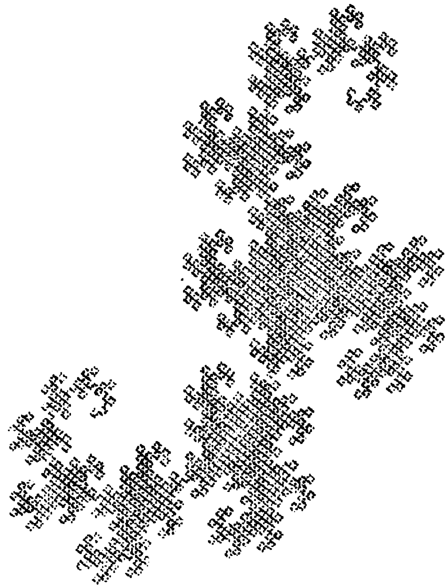
這個程式（８）在執行後，要等候大約３０秒才有動畫顯示，它會由左至右彈出一個球，由高至低，祇要稍強改良，便可以用作物理運動模擬程式。

龍的圖

一個很有中國構圖色彩程式，利用小方格的組合出一個圖案；你共有由１至７共七個大小可供繪畫顯示，見程式（９）

N角形

程式（１０）亦是最後一個程式，輸入要繪出的N角形之N數，程式便會繪出N角形及N角星了。



程式一

```
100  TEXT : HOME
110 A$ = "APPLE USER #21"
120  FOR X = 12 TO 23
130  HTAB X: VTAB X: PRINT A$
140  HTAB 24 - X: VTAB X: PRINT A
     $
150  HTAB X: VTAB 24 - X: PRINT A
     $
160  HTAB 24 - X: VTAB 24 - X: PRINT
     A$
170  NEXT
180  FOR D = 1 TO 24
190  FOR Y = 12 TO 23
200  VTAB D: HTAB Y: PRINT A$
210  VTAB D: HTAB 24 - Y: PRINT A
     $
220  NEXT : NEXT
```

程式二

```
100  TEXT : HOME
110 L = 1:A$ = "APPLE USER #21"
120  FOR A = 1 TO  LEN (A$) + 1
130  VTAB 22: HTAB 1: PRINT  CHR$
     (93);C$;
140 C$ =  LEFT$ (A$,L)
150  GET T$
160 L = L + 1: NEXT
```

程式三

```
100  FOR X = 0 TO 19
110  FOR Y = 0 TO 47
120 B =  PEEK ( - 16336):A =  SCRN(
     X,Y)
130  COLOR= A:A =  SCRN( 39 - X,Y
     )
140  PLOT 39 - X,Y: COLOR= A: PLOT
     X,Y
150  NEXT : NEXT
160  FOR Y = 0 TO 47
170  FOR X = 0 TO 19
180 B =  PEEK ( - 16336):A =  SCRN(
     X,Y)
190  COLOR= A:A =  SCRN( 39 - X,Y
     )
200  PLOT 39 - X,Y: COLOR= A: PLOT
     X,Y
210  NEXT : NEXT
```

程式四

```
100  INVERSE
110  FOR V = 23 TO 1 STEP  - 1
120  FOR H = 1 TO 40
130  VTAB V: HTAB H: PRINT " "
140  NEXT : NEXT
150  POKE 34,5: POKE 35,17
160  POKE 32,8: POKE 33,24
170  POKE 241,83
180  FOR Q = 1 TO 5
190  FOR S = 1 TO 24
200  HTAB S: PRINT " ": NEXT
210  FOR T = 24 TO 1 STEP  - 1
220  HTAB T: PRINT " ": NEXT : NEXT

230  HOME : TEXT
240  FOR K = 1 TO S
250  PRINT : NEXT : POKE 241,1: RUN
```

程式五

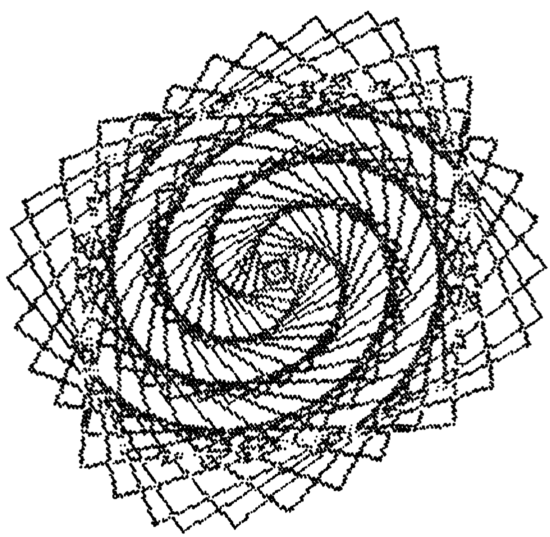
```
100  FOR I = 1 TO 1000
110  POKE  - 16299,0
120  HCOLOR= 3
130  POKE  - 16300,0
140  NEXT
```

程式六

```
100  FOR A = 769 TO 783
110  READ B: POKE A,B
120  NEXT
130  HGR : HCOLOR= 5
140  HPLOT 0,0
150  CALL 62454
160  HGR2
170  CALL 769
180  DATA  173,87,192,234,173,84,
     192,234
190  DATA  173,85,192,234,76,1,3
```

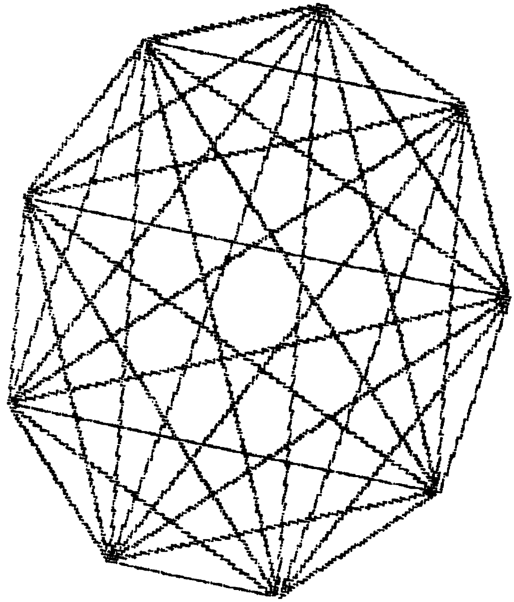
程式七

```
100  FOR I = 768 TO 777: READ P: POKE
     I,P: NEXT
110  POKE 232,0: POKE 233,3
120  HGR2 : HCOLOR= 3
130  FOR A = 1 TO 10000
140 S =  INT ( RND (1) * 5) + 1
150  FOR R = 1 TO 65 STEP S
160  ROT= R: SCALE= R: DRAW 1 AT
     139,95
170  NEXT
180  FOR R = 66 TO 130 STEP S
190  ROT= R: SCALE= 131 - R: XDRAW
     1 AT 139,95
200  NEXT : NEXT
210  DATA  1,0,4,0,33,63,54,45,4,
     0
```



程式八

```
100  HOME : HGR : HCOLOR= 3
110  DIM Y(600):A = 150:B = 270
120  HPLOT 0,A TO B,A
130  FOR I = 1 TO B
140 Y(I) = A -  ABS ( SIN (I / 9)
    ) *  EXP ( - I / 80) * A:Y(I
     + A) = A
150  NEXT
160  FOR F = 1 TO B
170  FOR I = 1 TO B
180  FOR J = 0 TO 1
190  HCOLOR= J * 3: HPLOT I + T *
    J,Y(I + T * J)
200  NEXT J,I
210 T = T + 1:T = T * (T < 9): NEXT
```



程式九

```
100  INPUT "SIZE=";N: TEXT
110  IF N > 0 AND N < 8 THEN 130
120  END
130 F = 128:E = 0:K = 4 ^ N
140  FOR X = 1 TO N:U = F:F =  -
    E / 2:E = U / 2: NEXT
150  HGR : HCOLOR= 3:X = 80:Y = 7
    0: HPLOT X,Y
160  FOR M = 1 TO K:Z = M:X = X +
    F:Y = Y +  + E: HPLOT  TO X,
    Y
170  FOR A = 0 TO 1:Z = Z / 2:Q =
     INT (Z):A = Z - Q: NEXT
180 A = (Q / 2 =  INT (Q / 2)) *
    2 - 1:U = F:F = A * E:E =  -
    A * U: NEXT
190  GOTO 100
```

程式十

```
100  TEXT : HOME : INPUT "NUMBER
    OF POINTS: ";A
110  HCOLOR= 3: HGR2
120  FOR X = 0 TO (2 * 3.14) - 0.
    001 STEP (2 * 3.14) / A
130  FOR Y = X + (2 * 3.14) / A TO
    (2 * 3.14) - .001 STEP (2 *
    3.14) / A
140  HPLOT ( COS (X) * 95 + 127),
    ( SIN (X) * 95 + 95) TO ( COS
    (Y) * 95 + 127),( SIN (Y) *
    95 + 95)
150  NEXT : NEXT
160  GET A$: RUN
```

魔方陣

這是一個高解像度遊戲，其玩法與「電子世界爭霸戰」差不多，你要盡量避免與在畫面出現的物體碰到。

整個畫面都是由方塊組成，你控制的方塊在畫面的正中，移動鍵包括：

←	左移
→	右移
A	上移
Z	下移

而經過的地方都會留下方塊，此外方塊亦會不斷地出現，不論任何情形之下，一碰到方塊便GAME O—VER 。

程式一

```
100  HGR : HOME
110  HCOLOR= 3
120  FOR I = 0 TO 6
130 : READ A: POKE 768 + I,A
140  NEXT
150  POKE 232,0: POKE 233,3
160  ROT= 0: SCALE= 5
170  FOR I = 4 TO 278 STEP 6
180 : DRAW 1 AT I,7
190 : DRAW 1 AT I,151
200  NEXT
210  FOR I = 13 TO 145 STEP 6
220 : DRAW 1 AT 4,I
230 : DRAW 1 AT 274,I
240  NEXT
250 K = 49152:X = 136:Y = 79
260  DATA 1,0,4,0,37,55,0
270  POKE 234,0:S = 0
280 N = X:O = Y:T =  PEEK (K)
290 X = X +  - 6 * (T = 136) + 6 *
     (T = 149)
300 Y = Y +  - 6 * (T = 193) + 6 *
     (T = 218)
310 P =  INT ( RND (1) * 45 + 1) *
     6 + 4
320 Q =  INT ( RND (1) * 23 + 1) *
     6 + 7
330  POKE 234,0
340  DRAW 1 AT X,Y
350 V =  PEEK (234)
360  DRAW 1 AT P,Q
370  IF (N = X AND Y = O) OR V <
     20 THEN S = S + (N <  > X OR
     O <  > Y): GOTO 390
380  TEXT : HOME : PRINT "SCORE:
     "S: END
390  VTAB 23: PRINT "SCORE: "S: GOTO
     280
```

■

內海渡輪

這是一個文字顯示式遊戲，你負責在一艘渡輪上掌舵，由此岸到彼岸往返數次，把貨物由彼岸載回；然而內海中有很多戰時遺留下來的魚雷，順着水流向上冲，如果你的渡輪不幸碰到魚雷便立即玩完。畫面符號及其所代表的意思：

# # #	渡輪
O O O	貨物
———	航道
！ ！ ！	魚雷

渡輪之控制鍵爲：

←	向左
→	向右

程式一

```
100 S = 49200:K = S - 48:X = 2
110  TEXT : HOME
120  COLOR= 13: HLIN 0,39 AT 6
130  POKE 34,4
140  FOR E = 0 TO 1
150 :X = X + ( PEEK (K) = 149) -
     ( PEEK (K) = 136)
160 :X = X - (X > 35) + (X < 2)
170 : VTAB 2: PRINT
180 : PRINT T TAB( X + 1) MID$ ("
     000",1,3 * L) TAB( 40)5 - T -
     L;
190 : HTAB X: PRINT "-###-"
200 :D =  SCRN( X,8) +  SCRN( X +
     1,8) +  SCRN( X + 2,8)
210 :T = T + (L AND X = 2)
220 :M = M + 1
230 :L = (L OR X = 35) AND X > 2
240 : CALL  - 912
250 :C =  RND (1) < (T + 1) / 5
260 : COLOR= C: PLOT  RND (1) * 3
     0 + 5,46
270 :Z =  PEEK (S * C) -  PEEK (S
     * C)
280 :E = T = 5 OR D
290  NEXT
300  HOME : TEXT
310  VTAB 5
320 R =  INT (8.3 - M / 100)
330 R = R - (R < 0) * R
340  GET A$
350  PRINT  MID$ ("BARGE SUNK!SPE
     ED=" +  MID$ ("SLOWGOODFASTQ
     UIKWARP",1 + 4 * R,4),1 + 11
      *  NOT D,11)
360  END
```

滙編語言入門(上)



• 前言

要進一步了解及發揮一部微型電腦的功能，學習滙編語言是必要的，對於要與外界接觸，雖然是一般的高階語言都可以做到，但實際上是存在着很多問題，而與外部機器連接的軟件介面，通常都是用滙編語言寫成的。

另外滙編語言可以提高運作速度。使用滙語言寫成的程式，可以比一般高階語言快數至數十倍的速

度。但其程式編寫較爲困難，這又增加程式設計師負擔。

本文是爲初學者而設，簡釋由6502滙編語言至其應用爲止。

6505微處理器

6502微處理器是由美國的MOS Technology 推出，它是適用於微型電腦的微處理器；現在用於Apple Ⅱc的65C02是6502的CMOS 版本，這枚晶片增加一些新的命令，而且將未定義的編碼都作NOP看待。

如要明白6502命令的工作，除了參看有關的書籍外，最好在你的Apple Ⅱ上進行實驗，這樣會有更深切的了解。

6502記存器的結構

圖（1）所示的，是6502的內部記存器，它的結構非常簡單，初學者只需花些少時間便可掌握這枚微處理器。

圖中的記存器，所示的包括累加器A，指示記存器X和Y，3者都是8bit的記存器，內部記存器除了有8bit的之外，還有一16bit的程式計數器PC，它使到6502可以定址至64K—byte的記憶，而通常零頁區域（$0000～$00FF）的記憶的使用率較爲頻繁（參看圖（2））。

Stack指示器所使用的地址，則已定於地址$0100～$01FF區域，雖然祇有256b—ytes，但在實際使用上已沒有問題。

狀態記存器一般被簡稱爲P記存器，它是由7個旗號組成的，這些旗號的狀態，都受到大部份操作命令的影響。進位旗號一般都與加算命令、減算命令和比較命令一起使用；加算命令作進位記數，而減算和比較命令則將其用作借位記數。進位旗號可通過兩個命令：CLC和SEC，分別把該旗號設定爲0及1。

在6502進行累加器加1或減1的運算，進位旗號的設定一般如下：

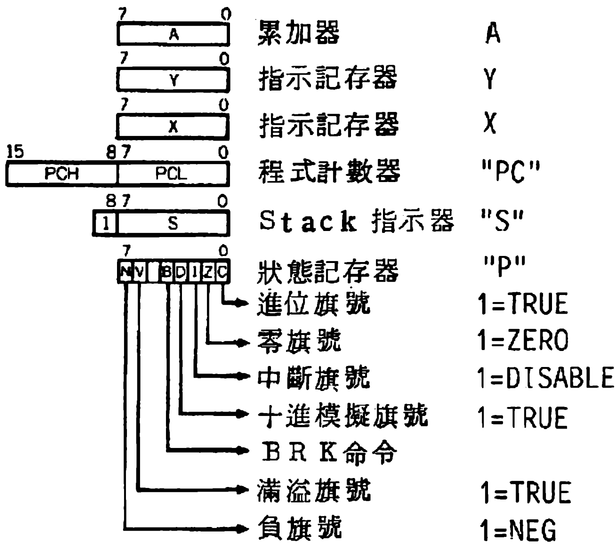
CLC（加算）　SEC（減算）

ADC#1　SBC#1

這是一般被使用的形式。

至於滿溢旗號，則是用來作加減算命令的進位及借位表示之用。CLV及SEV兩個命令分別是把其設定爲0及1。

另外一個十進模擬旗號，是把加減算命令，由2進數演算形式、變成10進數的演算形式，SED和CLD命令分別用作設定爲0及1。由於在電源啓動後6502的十進模擬旗號的設定不定，所以在使用時必須作初期的設定。



圖(1)6502的記存器

6502的定址法

在6502裏，利用它的8bit指示記存器，再加上多樣的定址法，使到6502這枚晶片的功能發揮到意料之外。

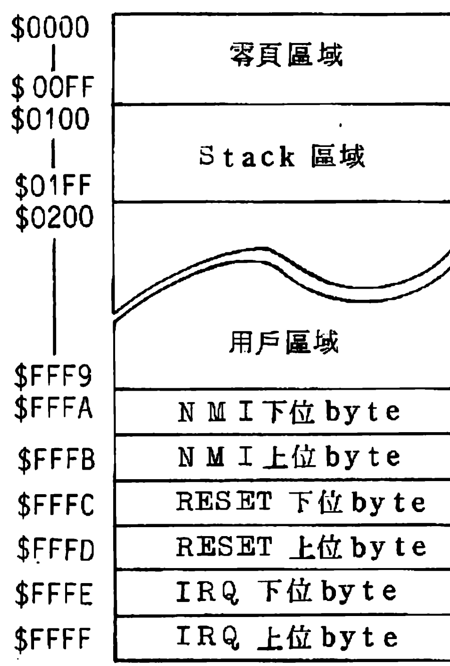
首先讓我們認識一下6502的命令種類，在圖(3)裏，我們可以看到3種6502命令的結構，而6502的定址法則有以下：

(1) 隱式定址法

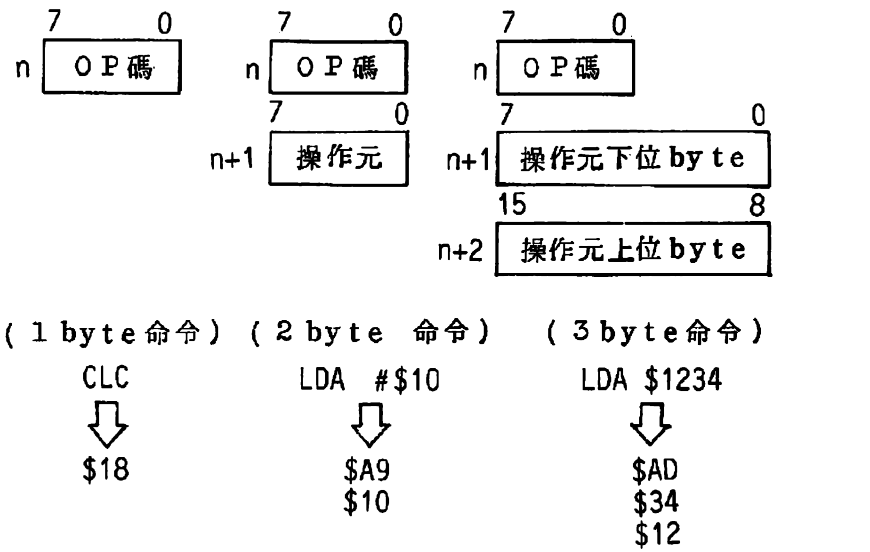
這個定址法的命令，祇有一個byte，並無其它參數。

(2) 直接定址法

這個定址法的命令，可用圖(3)的2 byte命令作爲例子，通常它有一個操作元的參數，這個定址法命令，一般都是對累加器來進行直



圖(2)6502被預定的地址空間



圖(3)6502機械語結構及例子

接的輸入、傳送、演算和比較等。

(3)絕對定址法

這個定址法是使用3 byte命令的，它是一個既基本且重要的定址法，命令後的兩個操作元參數是地址(見圖(4))。

△零頁定址法

這個定址法是絕對定址法的一個特例，由於它的上位地址byte是$00，所以把它省掉。也可說它是絕對定址法的短縮版本；一般在滙編語言的：

LDA$0000

都會被寫成：

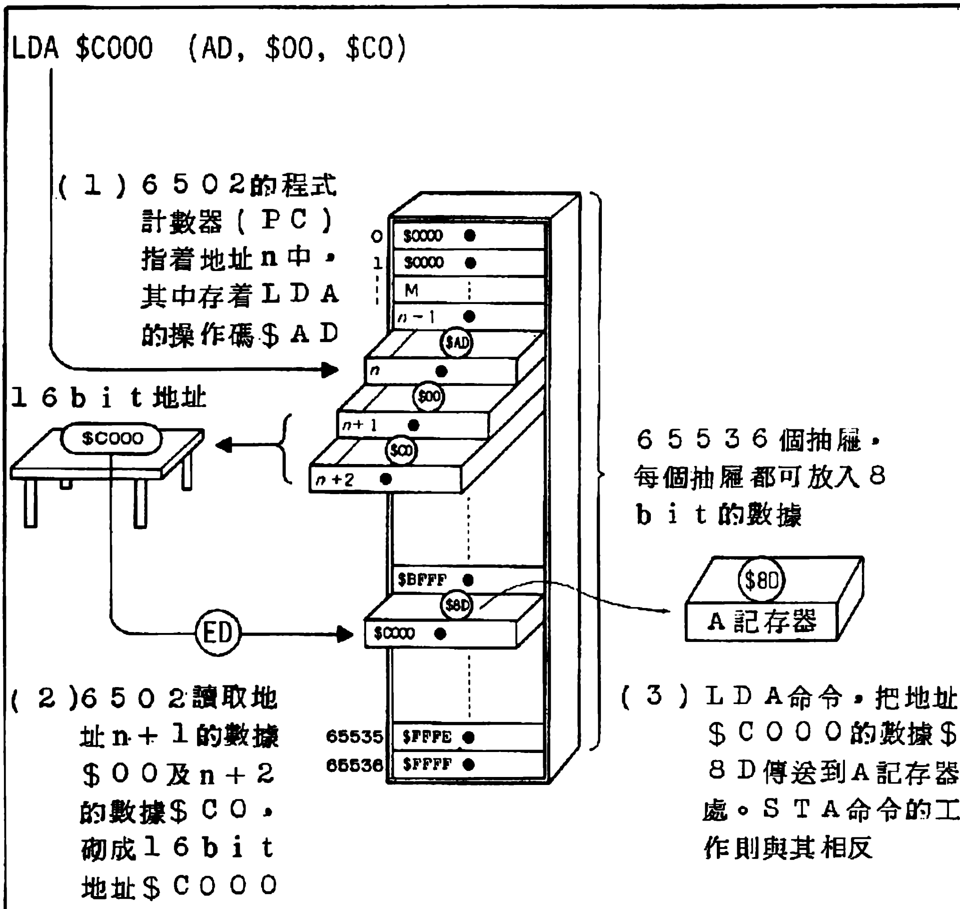
LDA$00

而且可以省掉一個byte。

(4)被指示記存器修飾的絕對定址法

[ABS，X ABS，Y]

它的形式與絕對定址法差不多，祇是把參數地址再加上X或Y指示記存器的內容，而成爲眞正的地址對象，我們稱這個眞正的地址爲實効地址，簡稱爲EA(Effe－ctive Address)，我們可以把它歸納爲以下的公式：

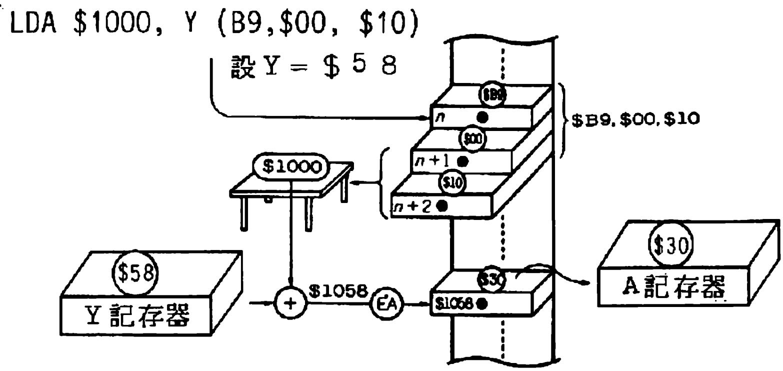


圖(4)絕對定址法中的LDA命令

EA＝16bit參數＋8
bit指示記存器
而由於指示記存器祇有8b－it，所以它的變化範圍將不會超過256.bytes（見圖（5））

（5）被指示記存器修飾的零頁定址法
[Z，PAGE，X　Z，P－AGE，Y]
這有定址法有如第（4）項一般，把地址加上指示記存器，變成EA，其公式爲：

EA＝[8bit參數＋8bit指示記存器] MOD $0100

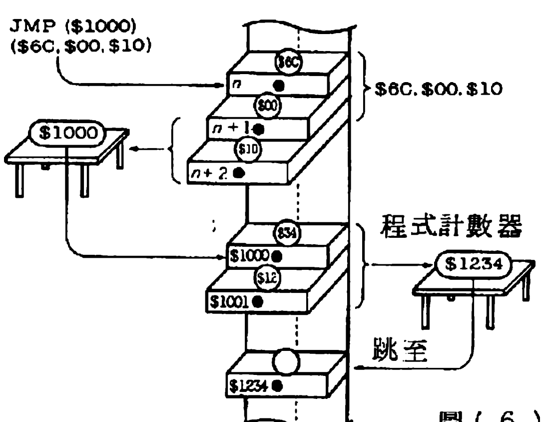


圖（５）ＡＢＳ，Ｘ　ＡＢＳ，Ｙ定址法

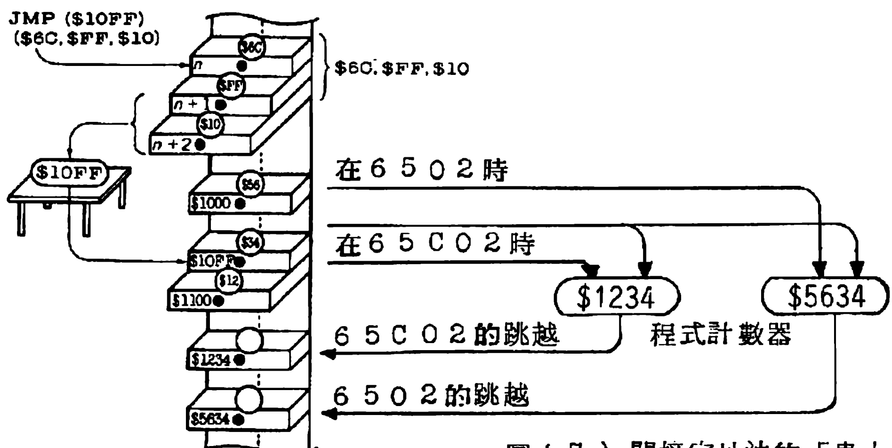
（６）間接定址法

這個定址法的例子命令爲Ｊ－ＭＰ（見圖（６）），這個定法把參數地址及其加１的地址，之兩個byte內容組合起來作爲眞正的實効地址。這個定址法在使用時會有些問題，這就是當地址是＄ＸＸＦＦ的＄ＦＦ爲尾時，在圖（７）中我們可以看到在ＣＰＵ６５０２時，它會指返同頁的地址＄ＸＸ００，所以在６５０２時，一般使用這個命令都會把它放在偶數中，以避產生這個「蟲」。

在新的ＣＭＯＳ　６５Ｃ０２ ＣＰＵ，則無這情況出現；



圖（６）間接定址法的ＪＭＰ命令



圖（7）間接定址法的「蟲」

這亦可以作爲6502及65C02判別的方法之一。

（7）相對定址法

6502裏共有8個分歧命令，這定址法定址範圍是−126至+129 bytes位置，實効地址的算法爲：

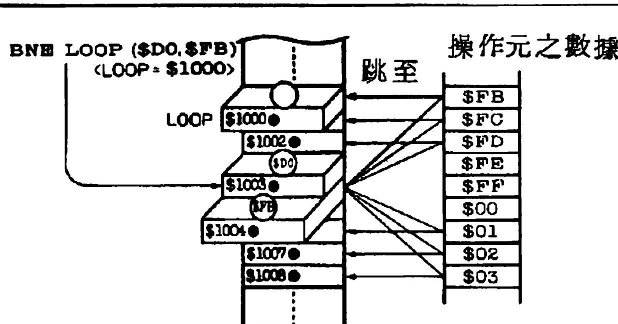
EA＝下一個命令地址＋（2補數的操作元參數）

在圖（8）裏，分歧命令BNE使用於地址$1003處，顯示其工作情況。

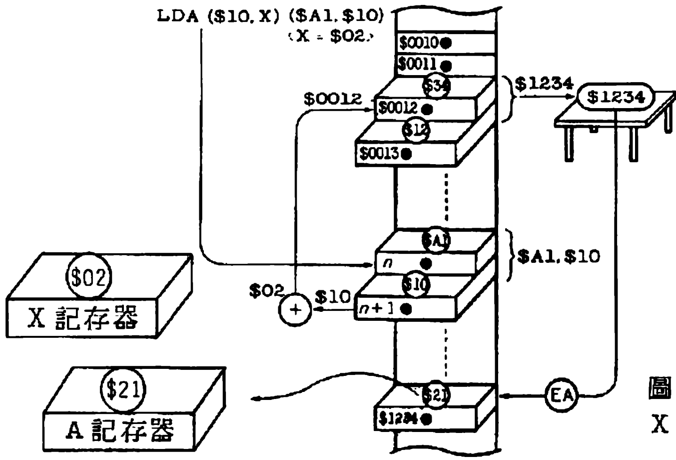
（8）被指示記存器修飾的間接定址法

（IND，X）

它使用的對象是零頁及X記存器，把操作元參數加上X記存器後的地址的內容，而成爲實効地址，情形就好像圖（9）般的例子。



圖（8）相對定址法



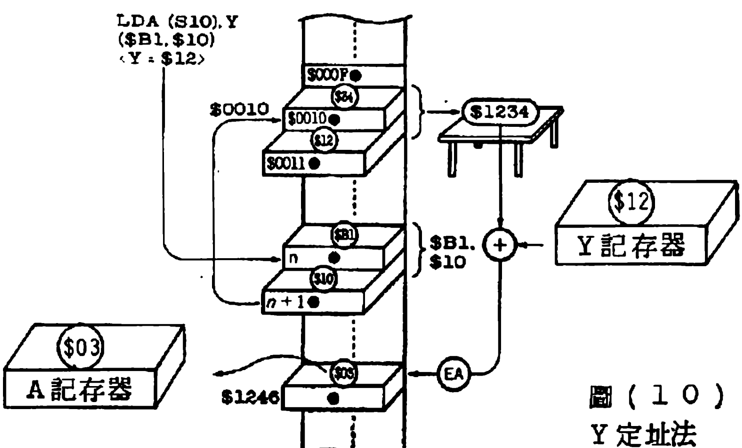
圖(9)(IND，X)定址法

(9)指示記存器修飾間接定址的定址法

(IND)，Y

這個定址法亦以零頁和Y記存器爲對象，工序首先把指示的操作元參數地址及其下個地址的內容，化作一個地址、而把這個新地址再加上Y記存器之內容，得出一個實效地址。在圖(10)中我們便有一個示範的例子。

在下期我們將會討論有關蘋果的監察器(Monitor)程式。 □



圖(10)(IND)，Y定址法

蘋 果 信 箱

敬啓者：

貴公司所出的刊物如微型電腦，現代電子以及蘋果用家月刊均有定期訂閱。內容頗充實，其中蘋果用家月刊，較爲價錢便宜，而內容可說是超值，而且受到我校同學的欣賞。本人是一位中學生，也是一位狂熱的C—omputer Fan，以零用錢的一部份用作購買貴出版社的刊物的資金。

在近期的蘋果用家月刊中發覺到，貴刊所新出的蘋果84專刊發行，於是拿了寶貴的$25去訂閱貴刊的刊物。

而且貴刊的84專刊，其中某部份的程式在運行上有阻誤而沒法解決，浪費了讀者寶貴的時間，電費及心機，好像在84專刊中一個名爲New Dos Programme中，在1ϕ7ϕ行中，運行出現illegal quautity error in1ϕ7ϕ，在請教Computer Fan 那麼高深研究的人也沒法解答。

可否請該Programme 的作者試解答一下其原因，是否電腦程式出現Problem 或是我的Computer 壞了？希望可以得到完善的解答。

電腦狂熱發燒友

上

電腦狂熱發燒讀者：

接過來信，立刻把原作者找出來，請他核對一下刊出的程式是否有錯，於是作者領命而去，而據作者回報，刊出在「蘋果84專刊」第4—46頁的NEW—DOS程式，的確有3點問題出現，這三點錯誤便是：—

1. 行270之SETUP 漏了REM一字。
2. 行530之HTAB指令後漏了H變數。
3. 行1030之PRINT D$指令後之等號爲多出來的。

上述三點改正後便可順利執行。

不過，上述各點與來信提及之illegal quautity error in 1070完全無關，所以對發燒讀者之問題未能提出解答，就此問題，便趁機爲大家介紹一下如何爲程式除錯。

首先要從了解指令開始，所有在BASIC 使用的指令其實都有一定的使用格式，而且必須要符合這指令的需要才可爲電腦所接受。而APPLE—SOFT 的指令其實不算多，故讀者應先行把各指令熟習其指令結構，這

樣便可改正，如上述之三項錯誤。

當在輸入程式時，要一心一意把全部程式輸入，千萬不可輸入了一半的程式試“RUN”一下，因程式未完全你不會有任何的結果，而且還可能因程式內的不完全部份損壞了程式結構，以致得不償失，好像程式中如有RESET或ERROR保護，而又在開始時有一GOSUB至後部份，當後半部未輸入時，如執行程式，可能會有HOLD機或程式損壞出現。

如要避免程式失去，最安全的方法的是邊打邊SAVE，不論在碟或帶上，均要一段時間後SAVE起全部程式，否則會有一些不能預知的事件發生（如停電？）

程式輸入後先核對一次，如有印字機則最理想不過，否則用80—C—OL來LIST亦比40COL好得多，全部程式核對過後，便可試R—UN一次，不過通常核對並不可把全部之錯誤找出，這時便要由實際執行中找出程式是否有錯。

大部份的ERROR均有行號指示，因此可由此句子找出該行之文法是否有錯，如指令的文法並沒有錯，則檢查其中的名變數值是否有錯，例如一數值不合規則（如POKE地址大於65536，POKE的DATA大於255，VTAB大於25………等等資料）。

如程式無法顯示出錯之行號數，則可檢查程式內是否有“ON ERR GOTO ×××”或是POKE 1010，××，POKE 1012，××，POKE 1013，××”之類的指令，因這兩指令均會把出錯之地點隱藏，故要先除去才可，但不可全行刪去，只是在這些出現的地方刪去，但如這些指令是單獨在一行時，則要加以一REM，以保持全部程式的執行行號。

如這階段仍然無效，則可試行APPLESOFT的TRACE指令，這指令執行後，RUN程式時每到一行程式，便會印出該行的號碼，以便查知程式的去向。

雖然上述只是初步去把程式除錯的技巧，實在亦是全部的除錯技巧，因編幅所限，如有機會，會再爲各讀者介紹如何爲程式除錯。

蘋果信箱謹覆

錯誤一

```
270 SETUP
270  REM  SETUP
```

錯誤二

```
530  VTAB V: HTAB R: GET X$: IF X$ = R$ THEN
      VTAB LV: HTAB : NORMAL : PRINT N$: GOTO
     380
530  VTAB V: HTAB R: GET X$: IF X$ = R$ THEN
      VTAB LV: HTAB H: NORMAL : PRINT N$: GOTO
     380
```

錯誤三

```
1030  HTAB I: NORMAL : PRINT "     ": IF  PEEK
      (S) = 106 THEN  PRINT D$ = "UNLOCK"N$
1030  HTAB I: NORMAL : PRINT "     ": IF  PEEK
      (S) = 106 THEN  PRINT D$"UNLOCK"N$
```

編輯先生：

敬希指教以下之問題：

1）怎樣計算一個機械語言程式之地址及長度？

2）有甚麼工具程式可以將有程序的磁碟變成一張空碟（新碟）？

本人有少少提議，希望貴刊多刊多一些實用，工具的程式。

謝謝

初學者K．T．

上

讀者K．T．君：

要計算一個機械語程式的長度，只要把程式最尾一個地址減去開始之地址再加1，這便是程式的長度，而起始地址便是開始之地址，如存在磁碟上，可用BSAVE FILENAME，A＄×××，L＄ＹＹＹ，或磁帶之ＣＡＬＬ—１５１＜ＣＲ＞，×××．ＺＺＺＷ＜ＣＲ＞，去完成，其中之×××是程式之起始地址，ＹＹＹ是程式長度，ＺＺＺ則是程式結尾地址。

而只需用ＩＮＩＴ 指令便可把有資料的磁碟變成空白的。

蘋果信箱謹覆

編輯先生：

我最近在一些外國雜誌，及黃金商場看見有一些叫８０８８Ｃａｒｄ的東西。他們說可以令一部８ｂｉｔ的Ａｐｐｌｅ變爲一部１６ｂｉｔ的電腦。我現在想請教一下是否那張Ｃａｒｄ眞的有這樣功能及其功能是否能與ＩＢＭ相比？！又Ａｐｐｌｅ的ｄｒｉｖｅ是ｓｉｎｇｌｅ－ｓｉｄｅｄ，而ＩＢＭ的是ｄｏｕｂｌｅ－ｓｉｄｅｄ的，那麼它究竟怎樣操作及Ａｐｐｌｅ是否可以利用爲ＩＢＭ而設的擴展卡?!

又如果想利用一部１６ｂｉｔ電腦作爲商業用途，一部ＩＢＭ或其Ｃ－ｏｍｐｔｉａｂｌｅ的電腦比一部Ａｐｐｌｅ加上８０８８Ｃａｒｄ的取捨應如何抉擇？

多謝指教

電腦發燒友

上

讀者發燒友君：

其實８０８８卡和Ｚ８０卡或６８０９卡的性質是一樣，只是可讓蘋果執行該枚ＣＰＵ可執行的操作系統。但其功能是未必可完全取代爲該Ｃ－ＰＵ而設計的機器的。

現時Ｚ８０卡運行的ＣＰ／Ｍ程式，亦是和原蘋果格式不相同的，包括４０軌和３５軌及資料格式的不同，而現在ＣＰ／Ｍ程式亦可蘋果上執行無誤。

而８０８８則有些少不同，因ＩＢＭ　ＰＣ最令人吸引是其本身的多種介面及容量，而蘋果加上了８０８８ＣＰＵ亦只能達到部份之功能，不能有十足的相容能力，但如所需要的是一發展８０８８ＣＰＵ用的程式則已十分足夠，但並不需要使用ＰＣ的其他硬件。

而ＰＣ的磁碟格式亦是有多種的，如單面，雙面或雙密度，故ＣＰ／Ｍ－８６或ＭＳ－ＤＯＳ是可在蘋果磁碟機上動作的，正如ＣＰ／Ｍ的磁碟一樣情況。

而除了有８０８８卡可插在蘋果內，亦聞說有廠商會推出一種可用Ｐ－Ｃ格式的磁碟機及擴展卡的"墨盒"（BLACK BOX）出現，不過這樣和ＰＣ其實沒有甚麽分別的？

所以如要對兩者在商業用途上分出高下，是有一定困難的，因蘋果已有多年歷史，現成的軟件十分多，而ＰＣ則有高速度，高容量及新科技的優點，這要用者自行權衡何者重要了。

蘋果信箱謹覆

徵求稿件

本刊歡迎各讀者投稿，無論是硬件製作、自製程式、軟件評介等一律歡迎。在投稿時請注意以下事項：

1）來稿請用４００字原稿紙書寫，文字一般以不超過２０００字爲限。
2）如稿件附有照片，必須清晰；而繪圖則須清楚整潔；並應附有其說明文字。
3）如係程式稿件，請把程式儲在磁碟或卡式帶上，並附上程式印列一份。
4）翻譯稿件必須註明原文出處，最好附上原文之副本乙份，以便參考。
5）本刊對來稿有權作刪改，而稿件一經刊出即致薄酬。
6）投稿者請將作者之眞實姓名、聯絡地址、電話付上，以便聯絡。
7）切勿一稿兩投。

■投稿請寄

九龍中央郵政信箱７２０２７號
蘋果用家月刊收

封面請註明「投稿」二字

蘋果用家月刊

1985年 2月20日出版
第4卷 第21期
每月20日出版

總編輯：黃仕焯
主　編：李友強
技術編輯：莫世傑
美術編輯：曹秀芬
封面設計：鍾志國
出　版：電子科學出版社
通訊地址：九龍中央郵政信箱72027號
通訊電話：〔編輯部〕☎ 3-801608
〔訂閱部〕☎ 3-805238

定價港幣四元

蘋果用家月刊

訂閱通知書

為方便資料編排，請以英文正楷填寫。

訂戶編號：____________________

NAME: ____________________ **TEL:** ____________________

ADDRESS: ____________________

____

____

附上劃綫支票No. ____________________

☐ 6期　　☐ 12期

由第________期至第________期

支票收款人請寫 "I. E. MICRO"

郵寄地址：九龍中央郵局信箱72027號



	本港	東南亞、外埠 (平郵)	空郵 東南亞	空郵 外埠	掛號另加
HK $ 6期	27.00	39.00	43.00	48.00	24.00
12期	54.00	78.00	86.00	95.00	48.00
US $ 6期	———	5.00	6.00	7.00	4.00
12期		10.00	11.00	12.50	7.00

蘋果用家月刊

電子科學出版社

每册定價港幣四元